Springer

Handbook of Engineering Statistics

Pham
Editor

Springer

Springer Handbook of Engineering Statistics

Springer Handbooks provide a concise compilation of approved key information on methods of research, general principles, and functional relationships in physics and engineering. The world's leading experts in the fields of physics and engineering will be assigned by one or several renowned editors to write the chapters comprising each volume. The content is selected by these experts from Springer sources (books, journals, online content) and other systematic and approved recent publications of physical and technical information.

The volumes will be designed to be useful as readable desk reference books to give a fast and comprehensive overview and easy retrieval of essential reliable key information, including tables, graphs, and bibliographies. References to extensive sources are provided.

Springer Handbook of Engineering Statistics

Hoang Pham (Ed.)

With CD-ROM, 377 Figures and 204 Tables

Springer

Hoang Pham
Rutgers the State University of New Jersey
Piscataway, NJ 08854, USA

British Library Cataloguing in Publication Data

Springer Handbook of Engineering Statistics
1. Engineering - Statistical methods
I. Pham, Hoang
620'.0072 ISBN-13: 9781852338060
ISBN-10: 1852338067

Library of Congress Control Number: 2006920465

ISBN-10: 1-85233-806-7 e-ISBN: 1-84628-288-8
ISBN-13: 978-1-85233-806-0 Printed on acid free paper

Production and typesetting: LE-TeX GbR, Leipzig
Handbook coordinator: Dr. W. Skolaut, Heidelberg
Typography, layout and illustrations: schreiberVIS, Seeheim
Cover design: eStudio Calamar Steinen, Barcelona
Cover production: *design&production* GmbH, Heidelberg
Printing and binding: Stürtz GmbH, Würzburg
Printed in Germany

SPIN 10956779 100/3100/YL 5 4 3 2 1 0

for Michelle, Hoang Jr., and David

Preface

The Springer Handbook of Engineering Statistics, altogether 54 chapters, aims to provide a comprehensive state-of-the-art reference volume that covers both fundamental and theoretical work in the areas of engineering statistics including failure time models, accelerated life testing, incomplete data analysis, stochastic processes, Bayesian inferences, data collection, Bootstrap models, burn-in and screening, competing risk models, correlated data analysis, counting processes, proportional hazards regression, design of experiments, DNA sequence analysis, empirical Bayes, genetic algorithms, evolutionary model, generalized linear model, geometric process, life data analysis, logistic regression models, longitudinal data analysis, maintenance, data mining, six sigma, Martingale model, missing data, influential observations, multivariate analysis, multivariate failure model, nonparametric regression, DNA sequence evolution, system designs, optimization, random walks, partitioning methods, resampling method, financial engineering and risks, scan statistics, semiparametric model, smoothing and splines, step-stress life testing, statistical process control, statistical inferences, statistical design and diagnostics, process control and improvement, biological statistical models, sampling technique, survival model, time-series model, uniform experimental designs, among others.

The chapters in this handbook have outlined into six parts, each contains nine chapters except Part E and F, as follows:

Part A Fundamental Statistics and Its Applications
Part B Process Monitoring and Improvement
Part C Reliability Models and Survival Analysis
Part D Regression Methods and Data Mining
Part E Statistical Methods and Modeling
Part F Applications in Engineering Statistics

Prof. Hoang Pham

All the chapters are written by over 100 outstanding scholars in their fields of expertise. I am deeply indebted and wish to thank all of them for their contributions and cooperation. Thanks are also due to the Springer staff for their patience and editorial work. I hope that practitioners will find this Handbook useful when looking for solutions to practical problems; researchers, statisticians, scientists and engineers, teachers and students can use it for quick access to the background, recent research and trends, and most important references regarding certain topics, if not all, in the engineering statistics.

January 2006 Hoang Pham
Piscataway, New Jersey

List of Authors

Susan L. Albin
Rutgers University
Department of Industrial
and Systems Engineering
96 Frelinghuysen Road
Piscataway, NJ 08854, USA
e-mail: *salbin@rci.rutgers.edu*

Suprasad V. Amari
Senior Reliability Engineer
Relex Software Corporation
540 Pellis Road
Greensburg, PA 15601, USA
e-mail: *suprasad.amari@relex.com*

Y. Alp Aslandogan
The University of Texas at Arlington
Computer Science and Engineering
416 Yates St., 206 Nedderman Hall
Arlington, TX 76019-0015, USA
e-mail: *alp@cse.uta.edu*

Jun Bai
JP Morgan Chase
Card Services
DE1-1073, 301 Walnut Street
Wilmington, DE 19801, USA
e-mail: *jun.bai@comcast.net*

Jaiwook Baik
Korea National Open University
Department of Information Statistics
Jong Ro Gu, Dong Sung Dong 169
Seoul, South Korea
e-mail: *baik@knou.ac.kr*

Amit K. Bardhan
University of Delhi – South Campus
Department of Operational Research
Benito Juarez Road
New Delhi, 110021, India
e-mail: *amit@or.du.ac.in*

Anthony Bedford
Royal Melbourne Institute of Technology University
School of Mathematical and Geospatial Sciences
Bundoora East Campus, Plenty Rd
Bundoora, Victoria 3083, Australia
e-mail: *anthony.bedford@rmit.edu.au*

James Broberg
Royal Melbourne Institute of Technology University
School of Computer Science & Information
Technology
GPO Box 2476V
Melbourne, Victoria 3001, Australia

Michael Bulmer
University of Queensland
Department of Mathematics
Brisbane, Qld 4072, Australia
e-mail: *m.bulmer@uq.edu.au*

Zhibin Cao
Arizona State University
Computer Science & Engineering Department
PO Box 878809
Tempe, AZ 85287-8809, USA
e-mail: *zbcao@asu.edu*

Philippe Castagliola
Université de Nantes and IRCCyN UMR CNRS 6597
Institut Universitaire de Technologie de Nantes
Qualité Logistique Industrielle et Organisation
2 avenue du Professeur Jean Rouxel
BP 539-44475 Carquefou, France
e-mail: *philippe.castagliola@univ-nantes.fr*

Giovanni Celano
University of Catania
Dipartimento di Ingegneria Industriale
e Meccanica
Viale Andrea Doria 6
Catania, 95125, Italy
e-mail: *gcelano@diim.unict.it*

Ling-Yau Chan
The University of Hong Kong
Department of Industrial and Manufacturing Systems Engineering
Pokfulam Road
Hong Kong
e-mail: *plychan@hku.hk*

Ted Chang
University of Virginia
Department of Statistics
Kerchhof Hall, PO Box 400135
Charlottesville, VA 22904-4135, USA
e-mail: *tcc8v@virginia.edu*

Victoria Chen
University of Texas at Arlington
Industrial and Manufacturing Systems Engineering
Campus Box 19017
Arlington, TX 76019-0017, USA
e-mail: *vchen@uta.edu*

Yinong Chen
Arizona State University
Computer Science and Engineering Department
PO Box 878809
Tempe, AZ 85287-8809, USA
e-mail: *yinong.chen@asu.edu*

Peter Dimopoulos
Royal Melbourne Institute of Technology University
Computer Science and IT
376-392 Swanston Street
Melbourne, 3001, Australia
e-mail: *dimpet@cs.rmit.edu.au*

Fenghai Duan
Department of Preventive
and Societal Medicine
984350 Nebraska Medical Center
Omaha, NE 68198-4350, USA
e-mail: *fduan@unmc.edu*

Veronica Esaulova
Otto-von-Guericke-University Magdeburg
Department of Mathematics
Universitätsplatz 2
Magdeburg, 39016, Germany
e-mail: *veronica.esaulova@gmail.com*

Luis A. Escobar
Louisiana State University
Department of Experimental Statistics
159-A Agricultural Administration Bldg.
Baton Rouge, LA 70803, USA
e-mail: *luis@lsu.edu*

Chun Fan
Arizona State University
Computer Science & Engineering Department
PO Box 878809
Tempe, AZ 85287-8809, USA
e-mail: *fanchun@asu.edu*

Kai-Tai Fang
Hong Kong Baptist University
Department of Mathematics
Kowloon Tong, Hong Kong
e-mail: *ktfang@hkbu.edu.hk*

Qianmei Feng
University of Houston
Department of Industrial Engineering
E206 Engineering Bldg 2
Houston, TX 77204, USA
e-mail: *qfeng@central.uh.edu*

Emilio Ferrari
University of Bologna
Department of Industrial
and Mechanical Engineering (D.I.E.M.)
viale Risorgimento, 2
Bologna, 40136, Italy
e-mail: *emilio.ferrari@unibo.it*

Sergio Fichera
University of Catania
Department Industrial
and Mechanical Engineering
avenale Andrea Doria 6
Catania, 95125, Italy
e-mail: *sfichera@diim.unict.it*

Maxim Finkelstein
University of the Free State
Department of Mathematical Statistics
PO Box 339
Bloemfontein, 9300, South Africa
e-mail: *finkelm.sci@mail.uovs.ac.za*

Mitsuo Gen
Waseda University
Graduate School of Information,
Production & Systems
2-7 Hibikino, Wakamatsu-Ku
Kitakyushu, 808-0135, Japan
e-mail: *gen@waseda.jp*

Amrit L. Goel
Syracuse University
Department of Electrical Engineering
and Computer Science
Syracuse, NY 13244, USA
e-mail: *goel@ecs.syr.edu*

Thong N. Goh
National University of Singapore
Industrial and Systems Engineering Dept.
10 Kent Ridge Crescent
Singapore, 119260, Republic of Singapore
e-mail: *isegohtn@nus.edu.sg*

Raj K. Govindaraju
Massey University
Institute of Information Sciences
and Technology
Palmerston North, 5301, New Zealand
e-mail: *k.govindaraju@massey.ac.nz*

Xuming He
University of Illinois at Urbana-Champaign
Department of Statistics
725 S. Wright Street
Champaign, IL 61820, USA
e-mail: *x-he@uiuc.edu*

Chengcheng Hu
Harvard School of Public Health
Department of Biostatistics
655 Huntington Avenue
Boston, MA 02115, USA
e-mail: *cchu@hsph.harvard.edu*

Feifang Hu
University of Virginia
Department of Statistics
Charlottesville, VA 22904, USA
e-mail: *fh6e@virginia.edu*

Hai Huang
Intel Corp CH3-20
Component Automation Systems
5000 W. Chandler Blvd.
Chandler, AZ 85226, USA
e-mail: *hai.huang2@intel.com*

Jian Huang
University of Iowa
Department of Statistics and Actuarial Science
241 Schaeffler Hall
Iowa City, IA 52242, USA
e-mail: *jian-huang@uiowa.edu*

Tao Huang
Yale University, School of Medicine
Department of Epidemiology and Public Health
60 College Street
New Haven, CT 06520, USA
e-mail: *t.huang@yale.edu*

Wei Jiang
Stevens Institute of Technology
Department of Systems Engineering
and Engineering Management
Castle Point of Hudson
Hoboken, NJ 07030, USA
e-mail: *wjiang@stevens.edu*

Richard Johnson
University of Wisconsin – Madison
Department of Statistics
1300 University Avenue
Madison, WI 53706-1685, USA
e-mail: *rich@stat.wisc.edu*

Kailash C. Kapur
University of Washington
Industrial Engineering
Box 352650
Seattle, WA 98195-2650, USA
e-mail: *kkapur@u.washington.edu*

P. K. Kapur
University of Delhi
Department of Operational Research
Delhi, 110007, India
e-mail: *pkkapur@or.du.ac.in*

Kyungmee O. Kim
Konkuk University
Department of Industrial Engineering
1 Hwayang-dong, Gwangjin-gu
Seoul, 143-701, S. Korea
e-mail: *kyungmee@konkuk.ac.kr*

Taeho Kim
Korea Telecom
Strategic Planning Office
221 Jungja-dong, Bundang-ku
Sungnam, Kyonggi-do, 463-711, S. Korea
e-mail: *taehokim@kt.co.kr*

Way Kuo
University of Tennessee
Department of Electrical
and Computer Engineering
124 Perkins Hall
Knoxville, TN 37996-2100, USA
e-mail: *way@utk.edu*

Paul Kvam
Georgia Institute of Technology
School of Industrial and Systems Engineering
755 Ferst Drive
Atlanta, GA 30332-0205, USA
e-mail: *pkvam@isye.gatech.edu*

Chin-Diew Lai
Massey University
Institute of Information Sciences and Technology
Turitea Campus
Palmerston North, New Zealand
e-mail: *c.lai@massey.ac.nz*

Jae K. Lee
University of Virginia
Public Health Sciences
PO Box 800717
Charlottesville, VA 22908, USA
e-mail: *jaeklee@virginia.edu*

Kit-Nam F. Leung
City University of Hong Kong
Department of Management Sciences
Tat Chee Avenue
Kowloon Tong, Hong Kong
e-mail: *mscknleun@cityu.edu.hk*

Ruojia Li
Global Statistical Sciences
Lilly Corporate Center
DC 3844
Indianapolis, IN 46285, USA
e-mail: *liru@lilly.com*

Wenjian Li
Javelin Direct, Inc.
Marketing Science
7850 Belt Line Road
Irving, TX 75063, USA
e-mail: *wenjian.li@javelindirect.com*

Xiaoye Li
Yale University
Department of Applied Mathematics
300 George Street
New Heaven, CT 06511, USA
e-mail: *xiaoye.li@yale.edu*

Yi Li
Harvard University
Department of Biostatistics
44 Binney Street, M232
Boston, MA 02115, USA
e-mail: *yili@jimmy.harvard.edu*

Hojung Lim
Korea Electronics Technology Institute (KETI)
Ubiquitous Computing Research Center
68 Yatap-dong, Bundang-Gu
Seongnam-Si, Gyeonggi-Do 463-816, Korea
e-mail: *hlim@keti.re.kr*

Haiqun Lin
Yale University School of Medicine
Department of Epidemiology
and Public Health
60 College Street
New Haven, CT 06520, USA
e-mail: *haiqun.lin@yale.edu*

Nan Lin
Washington University in Saint Louis
Department of Mathematics
Campus Box 1146, One Brookings Drive
St. Louis, MO 63130, USA
e-mail: *nlin@math.wustl.edu*

Wei-Yin Loh
University of Wisconsin – Madison
Department of Statistics
1300 University Avenue
Madison, WI 53706, USA
e-mail: *loh@stat.wisc.edu*

Jye-Chyi Lu
The School of Industrial and Systems Engineering
Georgia Institute of Technology
765 Ferst Drive, Campus Box 0205
Atlanta, GA 30332, USA
e-mail: *jclu@isye.gatech.edu*

William Q. Meeker, Jr.
Iowa State University
Department of Statistics
304C Snedecor Hall
Ames, IA 50011-1210, USA
e-mail: *wqmeeker@iastate.edu*

Mirjam Moerbeek
Utrecht University
Department of Methodology and Statistics
PO Box 80140
Utrecht, 3508 TC, Netherlands
e-mail: *m.moerbeek@fss.uu.nl*

Terrence E. Murphy
Yale University School of Medicine
Department of Internal Medicine
1 Church St
New Haven, CT 06437, USA
e-mail: *terrence.murphy@yale.edu*

D.N. Pra Murthy
The University of Queensland
Division of Mechanical Engineering
Brisbane, QLD 4072, Australia
e-mail: *p.murthy@uq.edu.au*

H. N. Nagaraja
Ohio State University
Department of Statistics
404 Cockins Hall, 1958 Neil Avenue
Columbus, OH 43210-1247, USA
e-mail: *hnn@stat.ohio-state.edu*

Toshio Nakagawa
Aichi Institute of Technology
Department of Marketing and Information Systems
1247 Yachigusa, Yagusa-cho
Toyota, 470-0392, Japan
e-mail: *toshi-nakagawa@aitech.ac.jp*

Joseph Naus
Rutgers University
Department of Statistics
Hill Center for the Mathematical Sciences
Piscataway, NJ 08855, USA
e-mail: *naus@stat.rutgers.edu*

Harriet B. Nembhard
Pennsylvania State University
Harold and Inge Marcus Department of Industrial and Manufacturing Engineering
University Park, PA 16802, USA
e-mail: *hbn2@psu.edu*

Douglas Noe
University of Illinois at Urbana-Champaign
Department of Statistics
725 S. Wright St.
Champaign, IL 61820, USA
e-mail: *dnoe@uiuc.edu*

Arrigo Pareschi
University of Bologna
Department of Industrial and Mechanical Engineering (D.I.E.M.)
viale Risorgimento, 2
Bologna, 40136, Italy
e-mail: *arrigo.pareschi@unibo.it*

Francis Pascual
Washington State University
Department of Mathematics
PO Box 643113
Pullman, WA 99164-3113, USA
e-mail: *jpascual@math.wsu.edu*

Raymond A. Paul
C2 Policy
U.S. Department of Defense (DoD)
3400 20th Street NE
Washington, DC 20017, USA
e-mail: *raymond.paul@osd.mil*

Alessandro Persona
University of Padua
Department of Industrial
and Technology Management
Stradella S. Nicola, 3
Vicenza, 36100, Italy
e-mail: *alessandro.persona@unipd.it*

Daniel Peña
Universidad Carlos III de Madrid
Departamento de Estadistica
C/Madrid 126
Getafe (Madrid), 28903, Spain
e-mail: *daniel.pena@uc3m.es*

Hoang Pham
Rutgers University
Department of Industrial
and Systems Engineering
96 Frelinghuysen Road
Piscataway, NJ 08854, USA
e-mail: *hopham@rci.rutgers.edu*

John Quigley
University of Strathclyde
Department of Management Science
40 George Street
Glasgow, G1 1QE, Scotland
e-mail: *j.quigley@strath.ac.uk*

Alberto Regattieri
Bologna University
Department of Industrial
and Mechanical Engineering
viale Risorgimento, 2
Bologna, 40136, Italy
e-mail: *alberto.regattieri@mail.ing.unibo.it*

Miyoung Shin
Kyungpook National University
School of Electrical Engineering
and Computer Science
1370 Sankyuk-dong, Buk-gu
Daegu, 702-701, Republic of Korea
e-mail: *shinmy@knu.ac.kr*

Karl Sigman
Columbia University in the City of New York,
School of Engineering and Applied Science
Center for Applied Probability (CAP)
500 West 120th St., MC: 4704
New York, NY 10027, USA
e-mail: *karl.sigman@columbia.edu*

Loon C. Tang
National University of Singapore
Department of Industrial
and Systems Engineering
1, Engineering Drive 2
Singapore, 117576, Singapore
e-mail: *isetlc@nus.edu.sg*

Charles S. Tapiero
Polytechnic University
Technology Management
and Financial Engineering
Six MetroTech Center
Brooklyn, NY 11201, USA
e-mail: *ctapiero@poly.edu*

Zahir Tari
Royal Melbourne Institute of Technology University
School of Computer Science
and Information Technology
GPO Box 2476V
Melbourne, Victoria 3001, Australia
e-mail: *zahirt@cs.rmit.edu.au*

Xiaolin Teng
Time Warner Inc.
Research Department
135 W 50th Street, 751-E
New York, NY 10020, USA
e-mail: *xiaolin_teng@timeinc.com*

Wei-Tek Tsai
Arizona State University
Computer Science & Engineering Department
PO Box 878809
Tempe, AZ 85287-8809, USA
e-mail: *wtsai@asu.edu*

Kwok-Leung Tsui
Georgia Institute of Technology
School of Industrial and Systems Engineering
765 Ferst Drive
Atlanta, GA 30332, USA
e-mail: *ktsui@isye.gatech.edu*

Fugee Tsung
Hong Kong University of Science
and Technology
Department of Industrial Engineering
and Logistics Management
Clear Water Bay
Kowloon, Hong Kong
e-mail: *season@ust.hk*

Lesley Walls
University of Strathclyde
Department of Management Science
40 George Street
Glasgow, G1 1QE, Scotland
e-mail: *lesley.walls@strath.ac.uk*

Wei Wang
Dana-Farber Cancer Institute
Department of Biostatistics
and Computational Biology
44 Binney Street
Boston, MA 02115, USA
e-mail: *wwang@jimmy.harvard.edu*

Kenneth Williams
Yale University
Molecular Biophysics and Biochemistry
300 George Street, G005
New Haven, CT 06520, USA
e-mail: *kenneth.williams@yale.edu*

Richard J. Wilson
The University of Queensland
Department of Mathematics
Brisbane, 4072, Australia
e-mail: *rjw@maths.uq.edu.au*

Baolin Wu
University of Minnesota, School of Public Health
Division of Biostatistics
A460 Mayo Building,
MMC 303, 420 Delaware St SE
Minneapolis, MN 55455, USA
e-mail: *baolin@biostat.umn.edu*

Min Xie
National University of Singapore
Dept. of Industrial & Systems Engineering
Kent Ridge Crescent
Singapore, 119 260, Singapore
e-mail: *mxie@nus.edu.sg*

Chengjie Xiong
Washington University in St. Louis
Division of Biostatistics
660 South Euclid Avenue, Box 8067
St. Louis, MO 63110, USA
e-mail: *chengjie@wubios.wustl.edu*

Di Xu
Amercian Express
Dept. of Risk Management
and Decision Science
200 Vesey Street
New York, NY 10285, USA
e-mail: *di.w.xu@aexp.com*

Shigeru Yamada
Tottori University
Department of Social Systems Engineering
Minami, 4-101 Koyama
Tottori-shi, 680-8552, Japan
e-mail: *yamada@sse.tottori-u.ac.jp*

Jun Yan
University of Iowa
Department of Statistics and Actuarial Science
241 Shaeffer Hall
Iowa City, IA 52242, USA
e-mail: *jyan@stat.uiowa.edu*

Shang-Kuo Yang
Department of Mechanical Engineering
National ChinYi Institute of Technology
No. 35, Lane 215, Sec. 1, Jungshan Rd.
Taiping City, 411, Taiwan, R.O.C.
e-mail: *skyang@ncit.edu.tw*

Kai Yu
Washington University in St. Louis,
School of Medicine
Division of Biostatistics
Box 8067
St. Louis, MO 63110, USA
e-mail: *yuka@mail.nih.gov*

Weichuan Yu
Yale Center for Statistical Genomics
and Proteomics, Yale University
Department of Molecular Biophysics
and Biochemistry
300 George Street
New Haven, CT 06511, USA
e-mail: *weichuan.yu@yale.edu*

Panlop Zeephongsekul
Royal Melbourne Institute of Technology University
School of Mathematical and Geospatial Sciences
GPO Box 2467V
Melbourne, Victoria 3000, Australia
e-mail: *panlopz@rmit.edu.au*

Cun-Hui Zhang
Rutgers University
Department of Statistics
Hill Center, Busch Campus
Piscataway, NJ 08854, USA
e-mail: *czhang@stat.rutgers.edu*

Heping Zhang
Yale University School of Medicine
Department of Epidemiology and Public Health
60 College Street
New Haven, CT 06520-8034, USA
e-mail: *heping.zhang@yale.edu*

Hongyu Zhao
Yale University School of Medicine
Department of Epidemiology and Public Health
60 College Street
New Haven, CT 06520-8034, USA
e-mail: *hongyu.zhao@yale.edu*

Kejun Zhu
China University of Geosciences
School of Management
No. 388 Lumo Road
Wuhan, 430074, Peoples Republic of China
e-mail: *zhubingl@pubilc.wh.hb.cn*

Contents

List of Tables

Part A Fundamental Statistics and Its Applications

1 Basic Statistical Concepts

2 Statistical Reliability with Applications

3 Weibull Distributions and Their Applications

6 Prediction Intervals for Reliability Growth Models with Small Sample Sizes

9 Modeling and Analyzing Yield, Burn-In and Reliability for Semiconductor Manufacturing: Overview

Part B Process Monitoring and Improvement

11 Statistical Methods for Product and Process Improvement

12 Robust Optimization in Quality Engineering

13 Uniform Design and Its Industrial Applications

15 Chain Sampling

16 Some Statistical Models for the Monitoring of High-Quality Processes

17 Monitoring Process Variability Using EWMA

List of Abbreviations

A

ABC	approximated bootstrap confidence
ACK	acknowledgment
ADDT	accelerated destructive degradation tests
ADI	average inter-demand interval
ADT	accelerated degradation test
AF	acceleration factor
AGP	arithmetico-geometric process
ALM	accelerated life model
ALT	accelerated life testing
AMA	arithmetic moving-average
ANN	artificial neural networks
ANOVA	analysis of variations
AP	arithmetic process
APC	automatic process control
AQL	acceptable quality level
AQM	active queue management
AR	autoregressive process
ARI	adjusted Rand index
ARL	average run length
ARMA	autoregressive and moving average
ARMDT	accelerated repeated measures degradation tests
ARRSES	adaptive response rate single-exponential smoothing
ART	accelerated reliability
ASN	average sample number
ASQ	American Society for Quality
ATI	average total inspection
AUC	area under the receiver operating characteristics curve
AW	additive Winter

B

BIB	burn-in board
BIR	built-in reliability
BLAST	Berkeley lazy abstraction software verification tool
BLUP	best linear unbiased predictor
BM	binomial model
BVE	bivariate exponential

C

CART	classification and regression tree
CBFQ	credit-based fair queueing
CBQ	class-based queues
CCD	central composite design
CDF	cumulative distribution function
CE	classification error
CF	characteristic function
CFE	Cauchy functional equation
CFF	call for fire
CHAID	chi-square automatic interaction detection
CID	collision-induced dissociation
CIM	cluster-image map
CLT	central limit theorem
CM	corrective maintenance
CML	canonical maximum likelihood
CMW	combination warranty
CNM	customer needs mapping
COPQ	cost of poor quality
COT	cumulative sum of T
cPLP	capacitated plant location problem
CRC	cumulative results criterion
CRUISE	classification rule with unbiased interaction selection and estimation
CS-CQ	cycle stealing with central queue
CS-ID	cycle stealing with immediate dispatch
CSALT	constant-stress accelerated life test
CSS	conditional single-sampling
CTQ	critical-to-quality
CUSUM	cumulative sum
CV	coefficient of variance
CV	cross-validation
CVP	critical value pruning
CX	cycle crossover
Cdf	cumulative distribution function
Cuscore	cumulative score
Cusum	cumulative sum

D

DBI	dynamic burn-in
DBSCAN	Density-based clustering
DCCDI	define, customer concept, design, and implement
DES	double-exponential smoothing
df	degrees of freedom
DFM	design for manufacturability
DFR	decreasing failure rate
DFR	design for reliability
DFSS	design for Six Sigma
DFY	design for yield
DLBI	die-level burn-in
DLBT	die-level burn-in and testing
DM	Data mining
DMADV	define, measure, analyze, design and verify
DMAIC	define, measure, analyze, improve, and control

DMAICT	define, measure, analyze, improve, control and technology transfer
DOE	design of experiments
DP	dynamic programming
DP	design parameters
DPMO	defects per million opportunities
DQ	dual-queue
DQLT	dual queue length threshold
DRD	dynamic robust design
DRR	deficit round-robin
DSSP	dependent stage sampling plan
DUT	device under test
DWC	discounted warranty cost
DoD	Department of Defense

E

EBD	equivalent business days
EBP	error-based pruning
EDWC	expected discounted warranty cost
EF	estimating function
EM	expectation maximization
EOQ	economic order quantity
EOS	electrical-over-stress
EQL	expected quality loss
ES	exponential smoothing
ESC	expected scrap cost
ESD	electrostatic discharge
ETC	expected total cost
EWC	expected warranty cost
EWMA	exponentially weighted moving average
EWMAST	exponentially weighted moving average chart for stationary processes

F

FCFS	first-come first-served
FDR	false discovery rate
FIR	fast initial response
FMEA	failure modes and effects analysis
FR	failure rate
FR	functional requirements
FRPW	free repair warranty
FRW	free replacement warranty
FSI	fixed sampling interval
FSW	full-service warranty
FTP	file transfer protocol
FWER	family-wise error rate

G

GA	genetic algorithms
GAB	generator armature bars
GAM	generalized additive model
GAOT	genetic algorithm optimization toolbox
GEE	generalized estimating equation
GERT	graphical evaluation and review technique
GLM	general linear model
GLM	generalized linear model
GLMM	generalized linear mixed model
GLRT	generalized likelihood ratio test
GP	geometric process
GUIDE	generalized, unbiased interaction detection and estimation

H

HALT	highly accelerated life tests
HCF	highest class first
HDL	high-density lipoprotein
HEM	heterogeneous error model
HEM	hybrid evolutionary method
HLA/RTI	high level architecture/runtime infrastructure
HPP	homogeneous Poisson process
HR	human resource
HTTP	hypertext transfer protocol

I

IC	inspection cost
ICOV	identify, characterize, optimize, verify
IDOV	identify, design, optimize, validate
IETF	internet engineering task force
IFM	inference functions for margins
IFR	increasing failure rate
i.i.d.	of independent and identically distributed
iid	independent identically distributed
IM	improvement maintenance
IT	information technology

K

KDD	knowledge discovery in databases
KGD	known good dies
KNN	*k*-nearest neighbors

L

LAC	lack of anticipation condition
LCEM	linear cumulative exposure model
LCF	lowest class first
LCL	lower control limits
LDA	linear discriminant analysis
LED	light emitting device
LIFO	last-in first-out
LLF	least loaded first
LLP	log-linear process
LMP	lack-of-memory property
LOC	lines of code
LOF	lack-of-fit
LPE	local pooled error

LQL	limiting quality level
LR	logistic regression
LSL	lower specification limit
LTI	low-turnaround-index
LTP	linear transportation problem

M

MAD	mean absolute deviation
MAD	median absolute deviation
MAPE	mean absolute percentage error
MARS	multivariate adaptive regression splines
MART	multiple additive regression tree
MC/DC	modified condition/decision coverage
MCF	minimum-cost-flow problem
MCMB	Markov chain marginal bootstrap
MCNR	Monte Carlo Newton–Raphson
MCS	Monte Carlo simulation
MCUSUM	multivariate cumulative sum
MDMSP	multidimensional mixed sampling plans
MDS	multiple dependent (deferred) state
MEP	minimum error pruning
MEWMA	multivariate exponentially weighted moving average
MGF	moment generating function
MILP	mixed integer linear programming model
ML	maximum-likelihood
MLDT	mean logistic delay time
MLE	maximum likelihood estimation
MME	method of moment estimates
MMSE	minimum mean squared error
MOLAP	multidimensional OLAP
MPDQ	multiple-priority dual queues
MPP	marked point process
MPP	multistage process planning
MRL	mean residual life
MS	mass spectrometry
MSA	measurement system analysis
MSE	mean square errors
MST	minimum spanning tree
MTBF	mean time before failure
MTBR	mean time between replacement
MTEF	marginal testing effort function
mTP	multiobjective transportation problem
MTS	Mahalanobis–Taguchi system
MTTF	mean time to failure
MTTR	mean time to repair
MVN	multivariate normal
MW	multiplicative Winter
MiPP	misclassification penalized posterior

N

NBM	nonoverlapping batch means
NHPP	nonhomogeneous Poisson process
NLP	nonlinear programming
NN	nearest neighbor
NPC	nutritional prevention of cancer
NTB	nominal-the-best case
NUD	new, unique, and difficult

O

OBM	overlapping batch means
OC	operating characteristic
OLAP	online analytical processing
OX	order crossover

P

PAR	phased array radar
PCB	printed circuit board
PDF	probability density function
pdf	probability density function
PEP	pessimistic error pruning
pFDR	proposed positive FDR
PH	proportional hazards
PID	proportional-integral-derivative
PLBI	package-level burn-in
PM	preventive maintenance
PMC	probabilistic model-based clustering
PMX	partial-mapped crossover
POF	physics-of-failure
PQL	penalized quasi-likelihood
PRM	probabilistic rational model
PRW	pro-rata warranty
PV	process variable

Q

QCQP	quadratically constrained quadratic programming
QDA	quadratic discriminant analysis
QFD	quality function deployment
QML	qualified manufacturing line
QSS	quick-switching sampling
QUEST	quick, unbiased and efficient statistical tree
QoS	quality of service

R

RBF	radial basis function
RCL	rate conservation law
RCLW	repair-cost-limit warranty
RD	Robust design
RED	random early-detection queue
REP	reduced error pruning
RF	random forest
RGS	repetitive group sampling
RIO	RED in/out
RNLW	repair-number-limit warranty

RP	renewal process
RPC	remote procedure call
RPN	priority number
RPN	risk priority number
RSM	response surface method
RSM	response surface methodology
RSM	response surface models
RTLW	repair-time-limit warranty
RV	random variable

S

SA	simulated annealing
SAFT	scale-accelerated failure-time
SAM	significance analysis of microarray
SAR	split and recombine
SBI	steady-state or static burn-in
SCC	special-cause charts
SCFQ	as self-clocked fair queueing
SCM	supply-chain management
s.d.	standard deviation
SDLC	software development life cycle
SDP	semidefinite program
SE	standard errors
SEM	structural equation models
SES	single-exponential smoothing
SEV	standard smallest extreme value
SF	survival function
SIMEX	simulation extrapolation
SIPOC	suppliers, inputs, process, outputs and customer
SIRO	service in random order
SMD	surface-mount devices
SMT	surface-mount technology
SNR	signal-to-noise ratios
SOAP	simple object access protocol
SOF	special operations forces
SOM	self-organizing maps
SOM	self-organizing (feature) map
SPC	statistical process control
SQL	structured query language
SRGM	software reliability growth models
SRM	seasonal regression model
SSBB	Six Sigma black belts
SSE	sum of squared errors
SSM	surface-to-surface missile
STS	standardized time series
SVM	support vector machine
SoS	system of systems

T

TAAF	test, analyse and fix
TAES	forecasting time series data that have a linear trend
TCP	transmission control protocol
TCP/IP	transmission control protocol/internet protocol
TDBI	test during burn-in
TDF	temperature differential factor
TQM	total quality management
TS	tracking signal
TSP	traveling-salesman problem

U

UBM	unified batch mean
UCL	upper control limits
UML	unified modeling language
USL	upper specification limit

V

VOC	voice of customer
VSI	variable sampling intervals
VaR	value at risk

W

WBM	weighted batch mean
WLBI	wafer-level burn-in
WLBT	wafer-level burn-in and testing
WLR	wafer-level reliability
WPP	Weibull probability plot
WRED	weighted RED
WRR	weighted round-robin
WSDL	web services description language

X

XML	extensible markup language

Y

Y2K	year 2000

Part A Fundamen

Part A Fundamental Statistics and Its Applications

Part A provides the concepts of fundamental statistics and its applications. The first group of five chapters exposes the readers, including researchers, practitioners and students, to the elements of probability, statistical distributions and inference and their properties. This comprehensive text can be considered as a foundation for engineering statistics. The first chapter provides basic statistics-related concepts, including a review of the most common distribution functions and their properties, parameter-estimation methods and stochastic processes, including the Markov process, the renewal process, the quasi-renewal process, and the nonhomogeneous Poisson process. Chapter 2 discusses the basic concepts of engineering statistics and statistical inference, including the properties of lifetime distributions, maximum-likelihood estimation, the likelihood ratio test, data modeling and analysis, and system reliability analysis, followed by variations of the Weibull and other related distributions, parameter estimations and hypothesis testing, and their applications in engineering. Chapter 4 describes the basic concept of characterizing functions based on random samples from common univariate discrete and continuous distributions such as the normal, exponential, Poisson, and multivariate distributions, including the Marshall–Olkin bivariate exponential and multivariate normal distributions. Chapter 5 discusses two-dimensional approaches to failure modeling, with applications in reliability and maintenance such as minimal repair and imperfect repair, and compares this through applications with the one-dimensional case.

The following four chapters cover the basic concepts in engineering statistics in specific topics such as reliability growth, warranty, marked point processes and burn-in. Chapter 6 presents the derivation of the prediction intervals for the time to detect the next fault for a small sample size by combining the Bayesian and frequentist approaches. It also provides examples to explain the predictions of the models, as well as their strengths and weaknesses. Chapter 7 gives an overview of various existing warranty models and policies and a summary of the issues in quantitative warranty modeling such as warranty cost factors, warranty policies, the warranty cost of multicomponent systems, the benefits of warranties, and optimal warranty policy analysis. Chapter 8 discusses the concept of a random market point process and its properties, including two-sided market point processes, counting processes, conditional intensity, the Palm distribution, renewal processes, stationary sequences, and time-homogeneous Poisson processes, while Chapt. 9 focuses on the yield, multilevel burn-in and reliability modeling aspects for applications in semiconductor manufacturing, considering various infant-mortality issues with the increased complexity of integrated circuits during manufacturing processes.

1. Basic Statistical Concepts

This brief chapter presents some fundamental elements of engineering probability and statistics with which some readers are probably already familiar, but others may not be. Statistics is the study of how best one can describe and analyze the data and then draw conclusions or inferences based on the data available. The first section of this chapter begins with some basic definitions, including probability axioms, basic statistics and reliability measures.

The second section describes the most common distribution functions such as the binomial, Poisson, geometric, exponential, normal, log normal, Student's t, gamma, Pareto, Beta, Rayleigh, Cauchy, Weibull and Vtub-shaped hazard rate distributions, their applications and their use in engineering and applied statistics.

The third section describes statistical inference, including parameter estimation and confidence intervals. Statistical inference is the process by which information from sample data is used to draw conclusions about the population from which the sample was selected that hopefully represents the whole population. This discussion also introduces the maximum likelihood estimation (MLE) method, the method of moments, MLE with censored data, the statistical change-point estimation method, nonparametic tolerance limits, sequential sampling and Bayesian methods.

The fourth section briefly discusses stochastic processes, including Markov processes, Poisson processes, renewal processes, quasi-renewal processes, and nonhomogeneous Poisson processes.

Finally, the last section provides a short list of books for readers who are interested in advanced engineering and applied statistics.

1.1 Basic Probability Measures

We start off this chapter by defining several useful terms:

1. Outcome: A result or observation from an experiment, which cannot be predicted with certainty.
2. Event: Subset of a set of all possible outcomes.
3. Probability: The relative frequency at which an event occurs in a large number of identical experiments.
4. Random variable: A function which assigns real numbers to the outcomes of an experiment.

5. Statistics: A function (itself a random variable) of one or more random variables, that does not depend upon any unknown parameters.

1.1.1 Probability Axioms

Now let C be a subset of the sample space ($C \subset \mathbb{Z}$). A probability set function, denoted by P(C), has the following properties:

1. $P(\mathbb{Z}) = 1, \quad P(C) \geq 0$
2. $P(C_1 \cup C_2 \cup \dots) = P(C_1) + P(C_2) + \dots$

where the subsets C_i have no elements in common (i. e., they are mutually exclusive).

Let C_1 and C_2 be two subsets of the sample space $\mathbb{Z}$. The conditional probability of getting an outcome in C_2 given that an outcome from C_1 is given by

$$P(C_2/C_1) = \frac{P(C_2 \cap C_1)}{P(C_1)}.$$

Let $C_1, C_2, \dots, C_n$ be n mutually disjoint subsets of the sample space $\mathbb{Z}$. Let C be a subset of the union of the C_is; that is

$$C \subset \bigcup_{i=1}^{n} C_i.$$

Then

$$P(C) = \sum_{i=1}^{n} P(C/C_i)P(C_i) \tag{1.1}$$

and

$$P(C_i/C) = \frac{P(C/C_i)P(C_i)}{\sum_{i=1}^{n} P(C/C_i)P(C_i)}.$$

Equation (1.1) is known as the law of total probability.

1.1.2 Basic Statistics

The cumulative distribution function (cdf) F is a unique function which gives the probability that a random variable X takes on values less than or equal to some value x. In other word, $F(x) = P(X \leq x)$.

The probability density function (pdf) f is the probability that X takes on the value x; that is, $f(x) = P(X = x)$.

The pdf satisfies the following two relations for discrete and continuous random variables, respectively,

$$\sum_{\text{all } x} f(x) = 1$$

and

$$\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 1.$$

In the continuous case, the pdf is the derivative of the cdf:

$$f(x) = \frac{\partial F(x)}{\partial x}.$$

The expected value of a random variable X is given by

$$E(X) = \sum_{\text{all } x} x f(x)$$

in the discrete case, and by

$$E(X) = \int_{-\infty}^{\infty} x f(x)\,\mathrm{d}x$$

in the continuous case. Similarly, the variance of a random variable X, denoted by σ^2, is a measure of how the values of X are spread about the mean value. It is defined as

$$\sigma^2 = E\,(X - \mu)^2\,.$$

It is calculated for discrete and continuous random variables, respectively, by

$$\sigma^2 = \sum_{\text{all } x} (x - \mu)^2 f(x)$$

and

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x)\,\mathrm{d}x.$$

The standard deviation of X, denoted by σ, is the square root of the variance.

The skewness coefficient of a random variable X is a measure of the symmetry of the distribution of X about its mean value μ, and is defined as

$$S_\mathrm{c} = \frac{E(X - \mu)^3}{\sigma^3}.$$

The skewness is zero for a symmetric distribution, negative for a left-tailed distribution, and positive for a right-tailed distribution.

Similarly, the kurtosis coefficient of a random variable X is a measure of how much of the mass of the distribution is contained in the tails, and is defined as

$$K_\mathrm{c} = \frac{E(X - \mu)^4}{\sigma^4}.$$

Obviously, kurtosis is always positive; however, larger values represent distributions with heavier tails.

Assume there are n random variables $X_1, X_2, \ldots, X_n$ which may or may not be mutually independent. The joint cdf, if it exists, is given by

$$P(X_1 \le x_1, X_2 \le x_2, \ldots X_n \le x_n) = \int_{-\infty}^{x_n} \int_{-\infty}^{x_{n-1}} .. \int_{-\infty}^{x_1} f(t_1, t_2, .., t_n)\,\mathrm{d}t_1\,\mathrm{d}t_2..\mathrm{d}t_n$$

If the n random variables are mutually statistically independent, then the joint pdf can be rewritten as

$$f(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(x_i).$$

The conditional distribution of a random variable Y given that another random variable X takes on a value x is given by:

$$f(y/X = x) = \frac{f(x, y)}{f_1(x)},$$

where

$$f_1(x) = \int_{-\infty}^{\infty} f(x, y)\,\mathrm{d}y.$$

Given a random sample of size n from a distribution, the sample mean and sample variance are defined as, respectively,

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

and

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \bar{X})^2.$$

1.1.3 Reliability Measures

Definitions of reliability given in the literature vary according to the practitioner or researcher. The generally accepted definition is as follows.

Definition 1.1

Reliability is the probability of success or the probability that the system will perform its intended function under specified design limits.

More specifically, reliability is the probability that a product or system will operate properly for a specified period of time (design life) under the design operating conditions (such as temperature, voltage, etc.) without failure. In other words, reliability can be used as a measure of the system's success at providing its function properly. Reliability is one of the quality characteristics that consumers require from manufacturers.

Mathematically, reliability $R(t)$ is the probability that a system will be successful in the interval from time 0 to time t:

$$R(t) = P(T > t), \quad t \ge 0, \tag{1.2}$$

where T is a random variable denoting the time-to-failure or failure time.

Unreliability, or the cdf $F(t)$, a measure of failure, is defined as the probability that the system will fail by time t.

$$F(t) = P(T \le t), \quad t \ge 0.$$

In other words, $F(t)$ is the failure distribution function. If the time-to-failure random variable T has a density function $f(t)$, then

$$R(t) = \int_{t}^{\infty} f(s)\,\mathrm{d}s$$

or, equivalently,

$$f(t) = -\frac{\mathrm{d}}{\mathrm{d}t}[R(t)].$$

The density function can be mathematically described in terms of T:

$$\lim_{\Delta t \to 0} P(t < T \le t + \Delta t).$$

This can be interpreted as the probability that the failure time T will occur between the operating time t and the next interval of operation $t + \Delta t$.

Consider a new and successfully tested system that operates well when put into service at time $t = 0$. The system becomes less likely to remain successful as the time interval increases. The probability of success for an infinite time interval is, of course, zero. Thus, the system starts to function at a probability of one and eventually decreases to a probability of zero. Clearly, reliability is a function of mission time. For example, one can say that the reliability of the system is 0.995 for a mission time of 24 h.

Example 1.1: A computer system has an exponential failure time density function

$$f(t) = \frac{1}{9000}\,\mathrm{e}^{-\frac{t}{9000}}, \quad t \geq 0.$$

The probability that the system will fail after the warranty (six months or 4380 h) and before the end of the first year (one year or 8760 h) is given by

$$P(4380 < T \leq 8760) = \int_{4380}^{8760} \frac{1}{9000}\,\mathrm{e}^{-\frac{t}{9000}}\,\mathrm{d}t = 0.237.$$

This indicates that the probability of failure during the interval from six months to one year is 23.7%.

Consider the Weibull distribution, where the failure time density function is given by

$$f(t) = \frac{\beta t^{\beta-1}}{\theta^{\beta}}\,\mathrm{e}^{-(\frac{t}{\theta})^{\beta}}, \quad t \geq 0,\ \theta > 0,\ \beta > 0.$$

Then the reliability function is

$$R(t) = \mathrm{e}^{-(\frac{t}{\theta})^{\beta}}, \quad t \geq 0.$$

Thus, given a particular failure time density function or failure time distribution function, the reliability function can be obtained directly. Section 1.2 provides further insight for specific distributions.

System Mean Time to Failure

Suppose that the reliability function for a system is given by $R(t)$. The expected failure time during which a component is expected to perform successfully, or the system mean time to failure (MTTF), is given by

$$\mathrm{MTTF} = \int_0^{\infty} t f(t)\,\mathrm{d}t \tag{1.3}$$

or, equivalently, that

$$\mathrm{MTTF} = \int_0^{\infty} R(t)\,\mathrm{d}t. \tag{1.4}$$

Thus, MTTF is the definite integral evaluation of the reliability function. In general, if $\lambda(t)$ is defined as the failure rate function, then, by definition, MTTF is not equal to $1/\lambda(t)$.

The MTTF should be used when the failure time distribution function is specified because the reliability level implied by the MTTF depends on the underlying failure time distribution. Although the MTTF measure is one of the most widely used reliability calculations, it is also one of the most misused calculations. It has been misinterpreted as a "guaranteed minimum lifetime". Consider the results given in Table 1.1 for a twelve-component life duration test.

A component MTTF of 3660 h was estimated using a basic averaging technique. However, one of the components failed after 920 h. Therefore, it is important to note that the system MTTF denotes the average time to failure. It is neither the failure time that could be expected 50% of the time nor is it the guaranteed minimum time of system failure, but mostly depends on the failure distribution.

A careful examination of (1.4) will show that two failure distributions can have the same MTTF and yet produce different reliability levels.

Table 1.1 Results from a twelve-component life duration test

Component	Time to failure (h)
1	4510
2	3690
3	3550
4	5280
5	2595
6	3690
7	920
8	3890
9	4320
10	4770
11	3955
12	2750

Failure Rate Function

The probability of a system failure in a given time interval $[t_1, t_2]$ can be expressed in terms of the reliability function as

$$\int_{t_1}^{t_2} f(t)\,\mathrm{d}t = \int_{t_1}^{\infty} f(t)\,\mathrm{d}t - \int_{t_2}^{\infty} f(t)\,\mathrm{d}t = R(t_1) - R(t_2)$$

or in terms of the failure distribution function (or the unreliability function) as

$$\int_{t_1}^{t_2} f(t)\,\mathrm{d}t = \int_{-\infty}^{t_2} f(t)\,\mathrm{d}t - \int_{-\infty}^{t_1} f(t)\,\mathrm{d}t = F(t_2) - F(t_1).$$

The rate at which failures occur in a certain time interval $[t_1, t_2]$ is called the *failure rate*. It is defined as the probability that a failure per unit time occurs in the interval, given that a failure has not occurred prior to t_1, the beginning of the interval. Thus, the failure rate is

$$\frac{R(t_1) - R(t_2)}{(t_2 - t_1)R(t_1)}.$$

Note that the failure rate is a function of time. If we redefine the interval as $[t, t + \Delta t]$, the above expression becomes

$$\frac{R(t) - R(t + \Delta t)}{\Delta t R(t)}.$$

The rate in the above definition is expressed in failures per unit time, but in reality the time units might instead correspond to miles, hours, trials, etc. The *hazard function* is defined as the limit of the failure rate as the interval approaches zero. Thus, the hazard function $h(t)$ is the instantaneous failure rate, and is defined by

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{R(t) - R(t + \Delta t)}{\Delta t R(t)} \\ &= \frac{1}{R(t)} \left[-\frac{\mathrm{d}}{\mathrm{d}t} R(t) \right] \\ &= \frac{f(t)}{R(t)}. \end{aligned} \tag{1.5}$$

The quantity $h(t)\,\mathrm{d}t$ represents the probability that a device of age t will fail in the small interval of time t to $(t + \mathrm{d}t)$. The importance of the hazard function is that it indicates the change in the failure rate over the life of a population of components by plotting their hazard functions on a single axis. For example, two designs may provide the same reliability at a specific point in time, but the failure rates up to this point in time can differ.

The death rate, in statistical theory, is analogous to the failure rate, as the nature of mortality is analogous to the hazard function. Therefore, the hazard function, hazard rate or failure rate function is the ratio of the pdf to the reliability function.

1.2 Common Probability Distribution Functions

This section presents some of the most common distribution functions and several hazard models that are applied in engineering statistics [1.1].

1.2.1 Discrete Random Variable Distributions

Binomial Distribution

The binomial distribution is one of the most widely used discrete random variable distributions in reliability and quality inspection. It has applications in reliability engineering, for example when one is dealing with a situation in which an event is either a success or a failure.

The binomial distribution can be used to model a random variable X which represents the number of successes (or failures) in n independent trials (these are referred to as Bernoulli trials), with the probability of success (or failure) being p in each trial. The pdf of the distribution is given by

$$P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x}, \; x = 0, 1, 2, \ldots, n,$$

$$\binom{n}{x} = \frac{n!}{x!(n - x)!},$$

where $n =$ number of trials, $x =$ number of successes, $p =$ single trial probability of success.

The mean of the binomial distribution is np and the variance is $np(1 - p)$. The coefficient of skewness is given by

$$S_c = \frac{1 - 2p}{\sqrt{np(1 - p)}}$$

and the coefficient of kurtosis is

$$K_c = 3 - \frac{6}{n} + \frac{1}{np(1 - p)}.$$

The reliability function $R(k)$ (i. e., at least k out of n items are good) is given by

$$R(k) = \sum_{x=k}^{n} \binom{n}{x} p^x (1 - p)^{n-x}.$$

Example 1.2: Suppose that, during the production of lightbulbs, 90% are found to be good. In a random sample of 20 lightbulbs, the probability of obtaining at least 18 good lightbulbs is given by

$$\begin{aligned} R(18) &= \sum_{x=18}^{20} \binom{20}{18} (0.9)^x (0.1)^{20-x} \\ &= 0.667. \end{aligned}$$

Poisson Distribution

Although the Poisson distribution can be used in a manner similar to the binomial distribution, it is used to deal with events in which the sample size is unknown. A Poisson random variable is a discrete random variable distribution with a probability density function given by

$$P(X = x) = \frac{\lambda^x \mathrm{e}^{-\lambda}}{x!} \quad \text{for } x = 0, 1, 2, \ldots \tag{1.6}$$

where λ = constant failure rate; x = is the number of events. In other words, $P(X = x)$ is the probability that exactly x failures occur.

A Poisson distribution is used to model a Poisson process. A Poisson random variable has a mean and a variance both equal to λ where λ is called the parameter of the distribution. The skewness coefficient is

$$S_\mathrm{c} = \frac{1}{\sqrt{\lambda}}$$

and the kurtosis coefficient is

$$K_\mathrm{c} = 3 + \frac{1}{\lambda}.$$

The Poisson distribution reliability up to time t, $R(k)$ (the probability of k or fewer failures), can be defined as follows

$$R(k) = \sum_{x=0}^{k} \frac{(\lambda t)^x \mathrm{e}^{-\lambda t}}{x!}.$$

This distribution can be used to determine the number of spares required for a system during a given mission.

Example 1.3: A nuclear plant is located in an area susceptible to both high winds and earthquakes. From historical data, the mean frequency of large earthquakes capable of damaging important plant structures is one every 50 y. The corresponding frequency of damaging high winds is once in 25 y. During a strong earthquake, the probability of structure damage is 0.1. During high winds, the damage probability is 0.05. Assume that earthquakes and high winds can be described by independent Poisson random variables and that the damage caused by these events are independent. Let us answer the following questions:

1. What is the probability of having strong winds but not large earthquakes during a 10y period?
2. What is the probability of having strong winds and large earthquakes in the 10y period?
3. What is the probability of building damage during the 10y period?

Considering the first question, let the random variables X and Y represent the number of earthquakes and the number of occurrences of high winds, respectively. We assume that the two random variables are statistically independent. The means of X and Y are, respectively, given by

$$\lambda_X = \frac{1}{50\,\mathrm{y}}\,(10\,\mathrm{y}) = 0.2$$

and

$$\lambda_Y = \frac{1}{25\,\mathrm{y}}\,(10\,\mathrm{y}) = 0.4\,.$$

The conditional damage probabilities are given as follows:

$$P(\text{damage/earthquake}) = 0.1$$

and

$$P(\text{damage/wind}) = 0.05.$$

Let event

A = {strong winds and no earthquakes},
B = {strong winds and large earthquakes},
C = {building damage}.

Assuming that the winds and earthquakes are independent of each other, the probability of having strong winds but not earthquakes during the 10 y period can be written as

$$\begin{aligned} P(A) &= P(\text{winds})P(\text{no earthquakes}) \\ &= [1 - P(\text{no winds})]P(\text{no earthquakes}) \end{aligned}$$

Therefore, we obtain

$$P(A) = (1 - \mathrm{e}^{-0.4})(\mathrm{e}^{-0.2}) = 0.27$$

For the second question, the probability of having strong winds and earthquakes during the 10 y period can be obtained from

$$\begin{aligned} P(B) &= P(\text{winds})P(\text{earthquakes}) \\ &= [1 - P(\text{no winds})][1 - P(\text{no earthquakes})] \\ &= (1 - \mathrm{e}^{-0.4})(1 - \mathrm{e}^{-0.2}) = 0.06\,. \end{aligned}$$

Finally, for the third question, we assume that multiple occurrences of earthquakes and high winds do not occur during the 10 y period. Therefore, the probability of

building damage can be written as

$$\begin{aligned} P(C) &= P(\text{damage/earthquakes})P(\text{earthquakes}) \\ &\quad + P(\text{damage/wind})P(\text{wind}) \\ &\quad - P(\text{damage/earthquakes and wind}) \\ &\quad P(\text{earthquake and wind}) \\ &= P(\text{damage/earthquakes})P(\text{earthquakes}) \\ &\quad + P(\text{damage/wind})P(\text{wind}) \\ &\quad - P(\text{damage/earthquakes})P(\text{damage/wind}) \\ &\quad P(\text{earthquake and wind}) \\ &= (1-\mathrm{e}^{-0.2})(0.1)+(1-\mathrm{e}^{-0.4})(0.05) \\ &\quad -(0.05)(0.1)(0.06) \\ &= 0.0343\,. \end{aligned}$$

Geometric Distribution

Consider a sequence of independent trials where each trial has the same probability of success, p. Let N be a random variable representing the number of trials until the first success. This distribution is called the geometric distribution. It has a pdf given by

$$P(N=n) = p\,(1-p)^{n-1}\,, \quad n = 1, 2, \ldots\,.$$

The corresponding cdf is

$$F(n) = 1-(1-p)^{n}\,, \quad n = 1, 2, \ldots\,.$$

The expected value and the variance are, respectively,

$$E(N) = \frac{1}{p}$$

and

$$V(N) = \frac{1-p}{p^2}.$$

Hypergeometric Distribution

The hypergeometric distribution is a discrete distribution that arises in sampling, for example. It has a pdf given by

$$f(x) = \frac{\binom{k}{x}\binom{N-k}{n-x}}{\binom{N}{n}} \quad x = 0, 1, 2, \ldots, n. \tag{1.7}$$

Typically, N will be the number of units in a finite population; n will be the number of samples drawn without replacement from N; k will be the number of failures in the population; and x will be the number of failures in the sample.

The expected value and variance of the hypergeometric random variable X are, respectively

$$E(X) = \frac{nk}{N}$$

and

$$V(X) = \frac{k(N-k)n(N-n)}{N^2(N-1)}.$$

1.2.2 Continuous Distributions

Exponential Distribution

The exponential distribution plays an essential role in reliability engineering because it has a constant failure rate. It has been used to model the lifetimes of electronic and electrical components and systems. This distribution is applicable to the case where a used component that has not failed is as good as a new component – a rather restrictive assumption. It should therefore be used carefully, since there are numerous situations where this assumption (known as the "memoryless property" of the distribution) is not valid.

If the time to failure is described by an exponential failure time density function, then

$$f(t) = \frac{1}{\theta}\,\mathrm{e}^{-\frac{t}{\theta}}, \quad t \geq 0,\ \theta > 0 \tag{1.8}$$

and this will lead to the reliability function

$$R(t) = \int_t^\infty \frac{1}{\theta}\,\mathrm{e}^{-\frac{s}{\theta}}\,\mathrm{d}s = \mathrm{e}^{-\frac{t}{\theta}}, \quad t \geq 0,$$

where $\theta = 1/\lambda > 0$ is an MTTF's parameter and $\lambda \geq 0$ is a constant failure rate.

The hazard function or failure rate for the exponential density function is constant, i. e.,

$$h(t) = \frac{f(t)}{R(t)} = \frac{\frac{1}{\theta}\,\mathrm{e}^{-\frac{1}{\theta}}}{\mathrm{e}^{-\frac{1}{\theta}}} = \frac{1}{\theta} = \lambda.$$

The failure rate for this distribution is λ, a constant, which is the main reason for this widely used distribution. Because of its constant failure rate, the exponential is an excellent model for the long flat "intrinsic failure" portion of the bathtub curve. Since most parts and systems spend most of their lifetimes in this portion of the bathtub curve, this justifies frequent use of the exponential distribution (when early failures or wearout is not a concern). The exponential model works well for interarrival times. When these events trigger failures, the exponential lifetime model can be used.

We will now discuss some properties of the exponential distribution that can be used to understand its characteristics and when and where it can be applied.

Property 1.1

(*Memoryless property*) The exponential distribution is the only continuous distribution that satisfies

$$P\{T \geq t\} = P\{T \geq t+s | T \geq s\} \quad \text{for } t > 0,\ s > 0. \tag{1.9}$$

This result indicates that the conditional reliability function for the lifetime of a component that has survived to time s is identical to that of a new component. This term is the so-called "used as good as new" assumption.

Property 1.2

If $T_1, T_2, \ldots, T_n$, are independently and identically distributed exponential random variables (r.v.'s) with a constant failure rate λ, then

$$2\lambda \sum_{i=1}^{n} T_i \sim \chi^2(2n), \tag{1.10}$$

where $\chi^2(2n)$ is a chi-squared distribution with $2n$ degrees of freedom. This result is useful for establishing a confidence interval for λ.

Uniform Distribution

Let X be a random variable with a uniform distribution over the interval (a, b) where $a < b$. The pdf is given by

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}.$$

The expected value and variance are, respectively,

$$E(X) = \frac{a+b}{2}$$

and

$$V(X) = \frac{(b-a)^2}{12}.$$

Normal Distribution

The normal distribution plays an important role in classical statistics due to the *Central Limit Theorem*. In production engineering, the normal distribution primarily applies to measurements of product susceptibility and external stress. This two-parameter distribution is used to describe mechanical systems in which a failure results from some wearout effect. The normal distribution takes the well-known bell shape. This distribution is symmetrical about the mean and the spread is measured by the variance. The larger the value, the flatter the distribution. The pdf is given by

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}, \quad -\infty < t < \infty,$$

where μ is the mean value and σ is the standard deviation. The cumulative distribution function (cdf) is

$$F(t) = \int_{-\infty}^{t} \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2} \mathrm{d}s.$$

The reliability function is

$$R(t) = \int_{t}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2} \mathrm{d}s.$$

There is no closed-form solution for the above equation. However, tables for the standard normal density function are readily available (see Table 1.6 in Sect. 1.A) and can be used to find probabilities for any normal distribution. If

$$Z = \frac{T-\mu}{\sigma}$$

is substituted into the normal pdf, we obtain

$$f(z) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{z^2}{2}}, \quad -\infty < Z < \infty.$$

This is a so-called standard normal pdf, with a mean value of 0 and a standard deviation of 1. The standardized cdf is given by

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}s^2} \mathrm{d}s, \tag{1.11}$$

where Φ is a standard normal distribution function. Thus, for a normal random variable T, with mean μ and standard deviation σ,

$$P(T \leq t) = P\left(Z \leq \frac{t-\mu}{\sigma}\right) = \Phi\left(\frac{t-\mu}{\sigma}\right),$$

where Φ yields the relationship required if standard normal tables are to be used.

It should be noted that the coefficent of kurtosis in the normal distribution is 3. The hazard function for a normal distribution is a monotonically increasing function

of t. This is easily shown by proving that $h'(t) \geq 0$ for all t. Since

$$h(t) = \frac{f(t)}{R(t)}$$

then

$$h'(t) = \frac{R(t)f'(t) + f^2(t)}{R^2(t)} \geq 0.$$

One can attempt this proof by using the basic definition of a normal density function f.

Example 1.4: A component has a normal distribution of failure times with $\mu = 2000\,\text{h}$ and $\sigma = 100\,\text{h}$. The reliability of the component at 1900 h is required.

Note that the reliability function is related to the standard normal deviate z by

$$R(t) = P\left(Z > \frac{t-\mu}{\sigma}\right),$$

where the distribution function for Z is given by (1.11). For this particular application,

$$\begin{aligned} R(1900) &= P\left(Z > \frac{1900-2000}{100}\right) \\ &= P(z > -1). \end{aligned}$$

From the standard normal table in Table 1.6 in Sect. 1.A, we obtain

$$R(1,900) = 1 - \Phi(-1) = 0.8413.$$

The value of the hazard function is found from the relationship

$$h(t) = \frac{f(t)}{R(t)} = \frac{\Phi\left(\frac{t-\mu}{\sigma}\right)}{\sigma R(t)},$$

where Φ is the pdf of the standard normal density. Here

$$\begin{aligned} h(1900) &= \frac{\Phi(-1.0)}{\sigma R(t)} = \frac{0.1587}{100(0.8413)} \\ &= 0.0019 \text{ failures/cycle}. \end{aligned}$$

The normal distribution is flexible enough to make it a very useful empirical model. It can be theoretical derived under assumptions matching many failure mechanisms. Some of these are: corrosion, migration, crack growth, and failures resulting from chemical reactions or processes in general. That does not mean that the normal distribution is always the correct model for these mechanisms, but it does perhaps explain why it has been empirically successful in so many of these cases.

Log Normal Distribution

The log normal lifetime distribution is a very flexible model that can empirically fit many types of failure data. This distribution, when applied in mechanical reliability engineering, is able to model failure probabilities of repairable systems, the compressive strength of concrete cubes, the tensile strength of fibers, and the uncertainty in failure rate information. The log normal density function is given by

$$f(t) = \frac{1}{\sigma t\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln t-\mu}{\sigma}\right)^2}, \quad t \geq 0, \tag{1.12}$$

where μ and σ are parameters such that $-\infty < \mu < \infty$, and $\sigma > 0$. Note that μ and σ are not the mean and standard deviations of the distribution.

Its relationship to the normal (just take natural logarithms of all of the data and time points and you have "normal" data) makes it easy to work with many good software analysis programs used to treat normal data.

Mathematically, if a random variable X is defined as $X = \ln T$, then X is normally distributed with a mean of μ and a variance of σ^2. That is,

$$E(X) = E(\ln T) = \mu$$

and

$$V(X) = V(\ln T) = \sigma^2.$$

Since $T = e^X$, the mean of the log normal distribution can be found via the normal distribution. Consider that

$$E(T) = E(e^X) = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\left[x - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]} dx.$$

By rearranging the exponent, this integral becomes

$$E(T) = e^{\mu+\frac{\sigma^2}{2}} \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}\left[x-(\mu+\sigma^2)\right]^2} dx.$$

Thus, the mean of the log normal distribution is

$$E(T) = e^{\mu+\frac{\sigma^2}{2}}.$$

Proceeding in a similar manner,

$$E(T^2) = E(e^{2X}) = e^{2(\mu+\sigma^2)}$$

so the variance for the log normal is

$$V(T) = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1).$$

The coefficient of skewness of this distribution is

$$S_c = \frac{e^{3\sigma^2} - 3e^{\sigma^2} + 2}{\left(e^{\sigma^2} - 1\right)^{\frac{3}{2}}}.$$

It is interesting that the skewness coefficient does not depend on μ and grows rapidly as the variance σ^2 increases.

The cumulative distribution function for the log normal is

$$F(t) = \int_0^t \frac{1}{\sigma s\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln s - \mu}{\sigma}\right)^2} ds$$

and this can be related to the standard normal deviate Z by

$$F(t) = P(T \le t) = P(\ln T \le \ln t) = P\left(Z \le \frac{\ln t - \mu}{\sigma}\right).$$

Therefore, the reliability function is given by

$$R(t) = P\left(Z > \frac{\ln t - \mu}{\sigma}\right) \tag{1.13}$$

and the hazard function would be

$$h(t) = \frac{f(t)}{R(t)} = \frac{\Phi\left(\frac{\ln t - \mu}{\sigma}\right)}{\sigma t R(t)}$$

where Φ is the cdf of standard normal density.

The log normal lifetime model, like the normal, is flexible enough to make it a very useful empirical model. It can be theoretically derived under assumptions matching many failure mechanisms, including corrosion, migration, crack growth and failures resulting from chemical reactions or processes in general. As with the normal distribution, this does not mean that the log normal is always the correct model for these mechanisms, but it suggests why it has been empirically successful in so many of these cases.

Student's *t* Distribution

Student's t probability density function of a random variable T is given by:

$$f(t) = \frac{\Gamma\left(\frac{r+1}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{r}{2}\right)\left(1 + \frac{t^2}{r}\right)^{\frac{r+1}{2}}} \quad \text{for } -\infty < t < \infty. \tag{1.14}$$

In other words, if a random variable T is defined as

$$T = \frac{W}{\sqrt{\frac{V}{r}}},$$

where W is a standard normal random variable and V has a chi-square distribution with r degrees of freedom, and W and V are statistically independent, then T is Student's t-distributed, and parameter r is referred to as the degrees of freedom (see Table 1.7 in Sect. 1.A).

The *F* Distribution

Let us define the random variable F is as follows

$$F = \frac{U/r_1}{V/r_2},$$

where U has a chi-square distribution with r_1 degrees of freedom, V is also chi-square-distributed, with r_2 degrees of freedom, and U and V are statistically independent, then the probability density function of F is given by

$$f(t) = \frac{\Gamma\left(\frac{r_1+r_2}{2}\right)\left(\frac{r_1}{r_2}\right)^{\frac{r_1}{2}}(t)^{\frac{r_1}{2}-1}}{\Gamma\left(\frac{r_1}{2}\right)\Gamma\left(\frac{r_2}{2}\right)\left(1 + \frac{r_1 t}{r_2}\right)^{\frac{r_1+r_2}{2}}} \quad \text{for } t > 0. \tag{1.15}$$

The F distribution is a two-parameter – r_1 and r_2 – distribution where the parameters are the degrees of freedom of the underlying chi-square random variables (see Table 1.8 in Sect. 1.A).

It is worth noting that if T is a random variable with a t distribution and r degrees of freedom, then the random variable T^2 has an F distribution with parameters $r_1 = 1$ and $r_2 = r$. Similarly, if F is F-distributed with r_1 and r_2 degrees of freedom, then the random variable Y, defined as

$$Y = \frac{r_1 F}{r_2 + r_1 F}$$

has a beta distribution with parameters $r_1/2$ and $r_2/2$.

Weibull Distribution

The exponential distribution is often limited in applicability owing to its memoryless property. The Weibull distribution [1.2] is a generalization of the exponential distribution and is commonly used to represent fatigue life, ball-bearing life and vacuum tube life. The Weibull distribution is extremely flexible and appropriate for modeling component lifetimes with fluctuating hazard rate functions and is used to represent various

types of engineering applications. The three-parameter probability density function is

$$f(t) = \frac{\beta(t-\gamma)^{\beta-1}}{\theta^{\beta}} \, \mathrm{e}^{-\left(\frac{t-\gamma}{\theta}\right)^{\beta}}, \quad t \geq \gamma \geq 0, \tag{1.16}$$

where θ and β are known as the scale and shape parameters, respectively, and γ is known as the location parameter. These parameters are always positive. By using different parameters, this distribution can follow the exponential distribution, the normal distribution, etc. It is clear that, for $t \geq \gamma$, the reliability function $R(t)$ is

$$R(t) = \mathrm{e}^{-\left(\frac{t-\gamma}{\theta}\right)^{\beta}} \quad \text{for } t > \gamma > 0,\ \beta > 0,\ \theta > 0 \tag{1.17}$$

hence,

$$h(t) = \frac{\beta(t-\gamma)^{\beta-1}}{\theta^{\beta}}, \quad t > \gamma > 0,\ \beta > 0,\ \theta > 0. \tag{1.18}$$

It can be shown that the hazard function decreases for $\beta < 1$, increases for $\beta > 1$, and is constant when $\beta = 1$.

Note that the Rayleigh and exponential distributions are special cases of the Weibull distribution at $\beta = 2$, $\gamma = 0$, and $\beta = 1$, $\gamma = 0$, respectively. For example, when $\beta = 1$ and $\gamma = 0$, the reliability of the Weibull distribution function in (1.17) reduces to

$$R(t) = \mathrm{e}^{-\frac{t}{\theta}}$$

and the hazard function given in (1.18) reduces to $1/\theta$, a constant. Thus, the exponential is a special case of the Weibull distribution. Similarly, when $\gamma = 0$ and $\beta = 2$, the Weibull probability density function becomes the Rayleigh density function. That is,

$$f(t) = \frac{2}{\theta} t \, \mathrm{e}^{-\frac{t^2}{\theta}} \quad \text{for } \theta > 0,\ t \geq 0.$$

Other Forms of Weibull Distributions

The Weibull distribution is widely used in engineering applications. It was originally proposed in order to represent breaking strength distributions of materials. The Weibull model is very flexible and has also been applied in many applications as a purely empirical model, with theoretical justification. Other forms of Weibull probability density function include, for example,

$$f(x) = \lambda\gamma x^{\gamma-1} \, \mathrm{e}^{-\lambda t^{\gamma}}. \tag{1.19}$$

When $\gamma = 2$, the density function becomes a Rayleigh distribution.

It is easy to show that the mean, variance and reliability of the above Weibull distribution are, respectively:

$$\begin{aligned} \text{Mean} &= \lambda^{\frac{1}{\gamma}} \Gamma\left(1 + \frac{1}{\gamma}\right); \\ \text{Variance} &= \lambda^{\frac{2}{\gamma}} \left\{ \Gamma\left(1 + \frac{2}{\gamma}\right) - \left[\Gamma\left(1 + \frac{1}{\gamma}\right)\right]^2 \right\}; \\ \text{Reliability} &= \mathrm{e}^{-\lambda t^{\gamma}}. \end{aligned} \tag{1.20}$$

Example 1.5: The time to failure of a part has a Weibull distribution with $\frac{1}{\lambda} = 250$ (measured in 10^5 cycles) and $\gamma = 2$. The part reliability at 10^6 cycles is given by:

$$R(10^6) = \mathrm{e}^{-(10)^2/250} = 0.6703.$$

The resulting reliability function is shown in Fig. 1.1.

Gamma Distribution

The gamma distribution can be used as a failure probability function for components whose distribution is skewed. The failure density function for a gamma distribution is

$$f(t) = \frac{t^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)} \mathrm{e}^{-\frac{t}{\beta}}, \quad t \geq 0,\ \alpha,\ \beta > 0, \tag{1.21}$$

where α is the shape parameter and β is the scale parameter. In this expression, $\Gamma(\alpha)$ is the gamma function, which is defined as

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} \mathrm{e}^{-t} \, \mathrm{d}t \quad \text{for } \alpha > 0.$$

Hence,

$$R(t) = \int_t^{\infty} \frac{1}{\beta^{\alpha}\Gamma(\alpha)} s^{\alpha-1} \mathrm{e}^{-\frac{s}{\beta}} \, \mathrm{d}s.$$

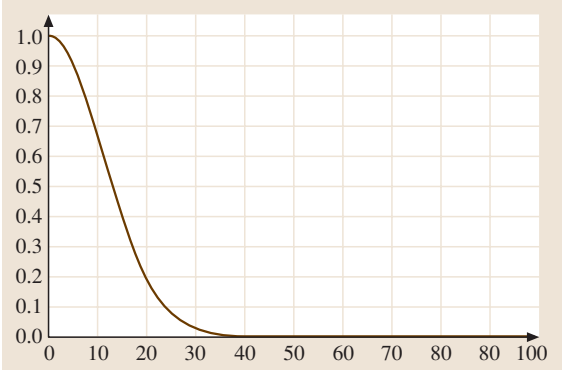


Fig. 1.1 Weibull reliability function versus time

If α is an integer, it can be shown by successive integration by parts that

$$R(t) = \mathrm{e}^{-\frac{t}{\beta}} \sum_{i=0}^{\alpha-1} \frac{\left(\frac{t}{\beta}\right)^i}{i!} \tag{1.22}$$

and

$$h(t) = \frac{f(t)}{R(t)} = \frac{\frac{1}{\beta^\alpha \Gamma(\alpha)} t^{\alpha-1} \mathrm{e}^{-\frac{t}{\beta}}}{\mathrm{e}^{-\frac{t}{\beta}} \sum_{i=0}^{\alpha-1} \frac{\left(\frac{t}{\beta}\right)^i}{i!}}. \tag{1.23}$$

The gamma density function has shapes that are very similar to the Weibull distribution. At $\alpha = 1$, the gamma distribution becomes the exponential distribution with a constant failure rate $1/\beta$. The gamma distribution can also be used to model the time to the nth failure of a system if the underlying failure distribution is exponential. Thus, if X_i is exponentially distributed with parameter $\theta = 1/\beta$, then $T = X_1 + X_2 + \cdots + X_n$ is gamma-distributed with parameters β and n.

Example 1.6: The time to failure of a component has a gamma distribution with $\alpha = 3$ and $\beta = 5$. Obtain the reliability of the component and the hazard rate at 10 time units.

Using (1.22), we compute

$$R(10) = \mathrm{e}^{-\frac{10}{5}} \sum_{i=0}^{2} \frac{\left(\frac{10}{5}\right)^i}{i!} = 0.6767\,.$$

The hazard rate is given by

$$h(10) = \frac{f(10)}{R(10)} = \frac{0.054}{0.6767} = 0.798 \text{ failures/unit time.}$$

The other form of the gamma probability density function can be written as follows:

$$f(x) = \frac{\beta^\alpha t^{\alpha-1}}{\Gamma(\alpha)} \mathrm{e}^{-t\beta}, \quad t > 0. \tag{1.24}$$

This pdf is characterized by two parameters: the shape parameter α and the scale parameter β. When $0 < \alpha < 1$, the failure rate monotonically decreases; when $\alpha > 1$, the failure rate monotonically increases; when $\alpha = 1$ the failure rate is constant.

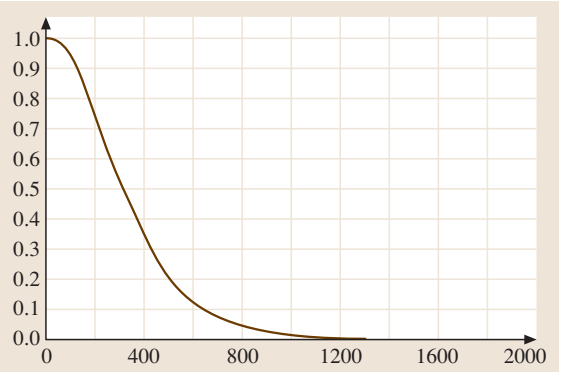


Fig. 1.2 Gamma reliability function versus time

The mean, variance and reliability of the gamma random variable are:

$$\text{Mean (MTTF)} = \frac{\alpha}{\beta};$$

$$\text{Variance} = \frac{\alpha}{\beta^2};$$

$$\text{Reliability} = \int_t^\infty \frac{\beta^\alpha x^{\alpha-1}}{\Gamma(\alpha)} \mathrm{e}^{-x\beta}\,\mathrm{d}x.$$

Example 1.7: A mechanical system time to failure is gamma-distributed with $\alpha = 3$ and $1/\beta = 120$. The system reliability at 280 h is given by

$$R(280) = \mathrm{e}^{\frac{-280}{120}} \sum_{k=0}^{2} \frac{\left(\frac{280}{120}\right)^2}{k!} = 0.851\,19$$

and the resulting reliability plot is shown in Fig. 1.2.

The gamma model is a flexible lifetime model that may offer a good fit to some sets of failure data. Although it is not widely used as a lifetime distribution model for common failure mechanisms, the gamma lifetime model is commonly used in Bayesian reliability applications.

Beta Distribution

The two-parameter beta density function $f(t)$ is given by

$$f(t) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} t^{\alpha-1}(1-t)^{\beta-1}\,,$$
$$0 < t < 1,\ \alpha > 0,\ \beta > 0\,, \tag{1.25}$$

where α and β are the distribution parameters. This two-parameter beta distribution is commonly used in many reliability engineering applications and also

plays an important role in the theory of statistics. Note that the beta-distributed random variable takes on values in the interval (0, 1), so the beta distribution is a natural model when the random variable represents a probability. Likewise, when $\alpha = \beta = 1$, the beta distribution reduces to a uniform distribution.

The mean and variance of the beta random variable are, respectively,

$$E(T) = \frac{\alpha}{\alpha + \beta}$$

and

$$V(T) = \frac{\alpha\beta}{(\alpha + \beta + 1)(\alpha + \beta)^2}.$$

Pareto Distribution

The Pareto distribution was originally developed to model income in a population. Phenomena such as city population size, stock price fluctuations and personal incomes have distributions with very long right tails. The probability density function of the Pareto distribution is given by

$$f(t) = \frac{\alpha k^\alpha}{t^{\alpha+1}}, \quad k \le t \le \infty. \tag{1.26}$$

The mean, variance and reliability of the Pareto distribution are:

$$\text{Mean} = k/(\alpha - 1) \quad \text{for } > 1;$$
$$\text{Variance} = \alpha k^2/[(\alpha - 1)^2(\alpha - 2)] \quad \text{for } \alpha > 2;$$
$$\text{Reliability} = \left(\frac{k}{t}\right)^\alpha.$$

The Pareto and log normal distributions are commonly used to model population size and economical incomes. The Pareto is used to fit the tail of the distribution, and the log normal is used to fit the rest of the distribution.

Rayleigh Distribution

The Rayleigh model is a flexible lifetime model that can apply to many degradation process failure modes. The Rayleigh probability density function is

$$f(t) = \frac{t}{\sigma^2} \exp\left(\frac{-t^2}{2\sigma^2}\right). \tag{1.27}$$

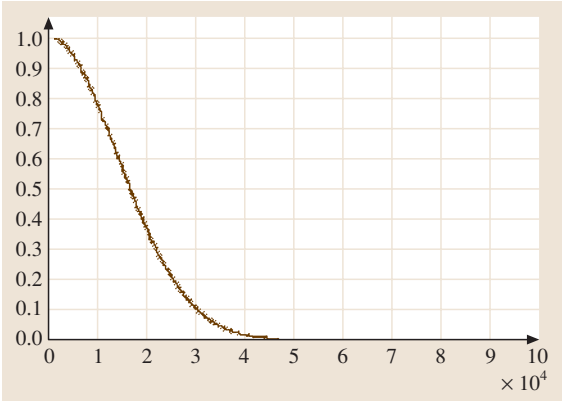


Fig. 1.3 Rayleigh reliability function versus time

The mean, variance and reliability of the Rayleigh function are:

$$\text{Mean} = \sigma\left(\frac{\pi}{2}\right)^{\frac{1}{2}};$$
$$\text{Variance} = \left(2 - \frac{\pi}{2}\right)\sigma^2;$$
$$\text{Reliability} = \mathrm{e}^{\frac{-\sigma t^2}{2}}.$$

Example 1.8: Rolling resistance is a measure of the energy lost by a tire under load when it resists the force opposing its direction of travel. In a typical car traveling at sixty miles per hour, about 20% of the engine power is used to overcome the rolling resistance of the tires. A tire manufacturer introduces a new material that, when added to the tire rubber compound, significantly improves the tire rolling resistance but increases the wear rate of the tire tread. Analysis of a laboratory test of 150 tires shows that the failure rate of the new tire increases linearly with time (h). This is expressed as

$$h(t) = 0.5 \times 10^{-8} t.$$

The reliability of the tire after one year (8760 h) of use is

$$R(1\text{ y}) = \mathrm{e}^{-\frac{0.5}{2} \times 10^{-8} \times (8760)^2} = 0.8254.$$

Figure 1.3 shows the resulting reliability function.

Vtub-Shaped Hazard Rate Distribution

Pham recently developed a two-parameter lifetime distribution with a Vtub-shaped hazard rate, known as

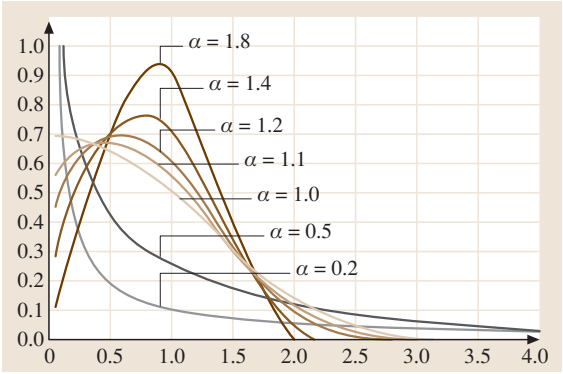


Fig. 1.4 Probability density function for various values of α with $a = 2$

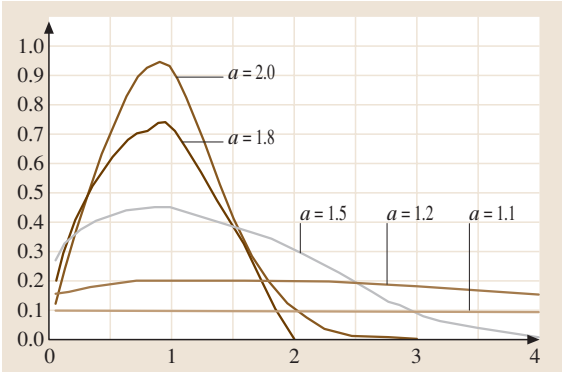


Fig. 1.5 Probability density function for various values of a with $\alpha = 1.5$

a *loglog distribution with a Vtub-shaped hazard rate* or a *Pham distribution* for short [1.3].

Note that the loglog distribution with a Vtub-shaped hazard rate and the Weibull distribution with bathtub-shaped failure rates are not the same. For the bathtub-shaped failure rate, after an initial "infant mortality period", the useful life of the system begins. During its useful life, the system fails at a constant rate. This period is then followed by a wearout period during which the system failure rate slowly increases with the onset of wearout. For the Vtub-shaped, after the infant mortality period, the system experiences a relatively low but increasing failure rate. The failure rate increases due to aging.

The Pham probability density function is given as follows [1.3]:

$$f(t) = \alpha \ln a t^{\alpha-1} a^{t^\alpha} \mathrm{e}^{1-a^{t^\alpha}}$$
$$\text{for } t > 0,\ a > 0,\ \alpha > 0. \tag{1.28}$$

The Pham distribution and reliability functions are

$$F(t) = \int_0^t f(x)\,\mathrm{d}x = 1 - \mathrm{e}^{1-a^{t^\alpha}}$$

and

$$R(t) = \mathrm{e}^{1-a^{t^\alpha}}, \tag{1.29}$$

respectively. The corresponding failure rate of the Pham distribution is given by

$$h(t) = \alpha \ln a t^{\alpha-1} a^{t^\alpha}. \tag{1.30}$$

Figures 1.4 and 1.5 describe the density functions and failure rate functions for various values of a and α.

Two-Parameter Hazard Rate Function

This is a two-parameter function that can have increasing and decreasing hazard rates. The hazard rate $h(t)$, the reliability function $R(t)$ and the pdf are, respectively, given as follows

$$h(t) = \frac{n\lambda t^{n-1}}{\lambda t^n + 1} \quad \text{for } n \geq 1,\ \lambda > 0,\ t \geq 0, \tag{1.31}$$
$$R(t) = \mathrm{e}^{-\ln(\lambda t^N + 1)}$$

and

$$f(t) = \frac{n\lambda t^{n-1}}{\lambda t^n + 1} \mathrm{e}^{-\ln(\lambda t^n + 1)}, \quad n \geq 1,\ \lambda > 0,\ t \geq 0,$$

where n = shape parameter; λ = scale parameter.

Three-Parameter Hazard Rate Function

This is a three-parameter distribution that can have increasing and decreasing hazard rates. The hazard rate $h(t)$ is given as

$$h(t) = \frac{\lambda(b+1)[\ln(\lambda t + \alpha)]^b}{(\lambda t + \alpha)},$$
$$b \geq 0,\ \lambda > 0,\ \alpha \geq 0,\ t \geq 0. \tag{1.32}$$

The reliability function $R(t)$ for $\alpha = 1$ is

$$R(t) = \mathrm{e}^{-[\ln(\lambda t + \alpha)]^{b+1}}.$$

The probability density function $f(t)$ is

$$f(t) = \mathrm{e}^{-[\ln(\lambda t + \alpha)]^{b+1}} \frac{\lambda(b+1)[\ln(\lambda t + \alpha)]^b}{(\lambda t + \alpha)},$$

where $b =$ shape parameter; $\lambda =$ scale parameter, and $\alpha =$ location parameter.

Extreme-Value Distribution

The extreme-value distribution can be used to model external events such as floods, tornadoes, hurricanes and high winds in risk applications. The cdf of this distribution is given by

$$F(t) = \mathrm{e}^{-\mathrm{e}^{y}} \quad \text{for } -\infty < t < \infty. \tag{1.33}$$

Cauchy Distribution

The Cauchy distribution can be applied when analyzing communication systems where two signals are received and one is interested in modeling the ratio of the two signals. The Cauchy probability density function is given by

$$f(t) = \frac{1}{\pi(1+t^2)} \quad \text{for } -\infty < t < \infty. \tag{1.34}$$

It is worth noting that the ratio of two standard normal random variables is a random variable with a Cauchy distribution.

1.3 Statistical Inference and Estimation

The problem of "point estimation" is that of estimating the parameters of a population, such as λ or θ from an exponential, μ and σ^2 from a normal, etc. It is assumed that the type of population distribution involved is known, but the distribution parameters are unknown and they must be estimated using collected failure data. This section is devoted to the theory of estimation and discusses several common estimation techniques, such as maximum likelihood, method of moments, least squared estimation, and Bayesian methods. We also discuss confidence interval and tolerance limit estimation. For example, assume that n independent samples are drawn from the exponential density function $f(x;\lambda) = \lambda\,\mathrm{e}^{-\lambda x}$ for $x > 0$ and $\lambda > 0$. Then the joint probability density function (pdf) or sample density (for short) is given by

$$f(x_1,\lambda)\cdot f(x_1,\lambda)\cdots f(x_1,\lambda) = \lambda^n\,\mathrm{e}^{-\lambda\sum_{i-1}^{n} x_i}. \tag{1.35}$$

The problem here is to find a "good" point estimate of λ, which is denoted by $\hat{\lambda}$. In other words, we want to find a function $h(X_1, X_2, \ldots, X_n)$ such that, if $x_1, x_2, \ldots, x_n$ are the observed experimental values of $X_1, X_2, \ldots, X_n$, the value $h(x_1, x_2, \ldots, x_n)$ will be a good point estimate of λ. By "good", we mean that it possesses the following properties:

- unbiasedness,
- consistency,
- efficiency (minimum variance),
- sufficiency.

In other words, if $\hat{\lambda}$ is a good point estimate of λ, then one can select a function $h(X_1, X_2, \ldots, X_n)$ where $h(X_1, X_2, \ldots, X_n)$ is an unbiased estimator of λ and the variance of $h(X_1, X_2, \ldots, X_n)$ is a minimum. We will now present the following definitions.

Unbiasedness. For a given positive integer n, the statistic $Y = h(X_1, X_2, \ldots, X_n)$ is called an unbiased estimator of the parameter θ if the expectation of Y is equal to a parameter θ; that is,

$$E(Y) = \theta.$$

Consistency. The statistic Y is called a consistent estimator of the parameter θ if Y converges stochastically to a parameter θ as n approaches infinity. If ε is an arbitrarily small positive number when Y is consistent, then

$$\lim_{n\to\infty} P(|Y-\theta| \le \varepsilon) = 1.$$

Minimum Variance. The statistic Y is called the minimum variance unbiased estimator of the parameter θ if Y is unbiased and the variance of Y is less than or equal to the variance of every other unbiased estimator of θ. An estimator that has the property of minimum variance in large samples is said to be efficient.

Sufficiency. The statistic Y is said to be sufficient for θ if the conditional distribution of X, given that $Y = y$, is independent of θ.

This is useful when determining a lower bound on the variance of all unbiased estimators. We now establish a lower bound inequality known as the *Cramér–Rao inequality*.

Cramér–Rao Inequality. Let $X_1, X_2, \ldots, X_n$ denote a random sample from a distribution with pdf $f(x;\theta)$ for $\theta_1 < \theta < \theta_2$, where θ_1 and θ_2 are known. Let $Y =$

$h(X_1, X_2, \ldots, X_n)$ be an unbiased estimator of θ. The lower bound inequality on the variance of Y, $\mathrm{Var}(Y)$, is given by

$$\mathrm{Var}(Y) \geq \frac{1}{nE\left[\left(\frac{\partial \ln f(x;\theta)}{\partial\theta}\right)^2\right]} = \frac{1}{-nE\left(\frac{\partial^2 \ln f(x;\theta)}{\partial\theta^2}\right)}. \tag{1.36}$$

Theorem 1.1

An estimator $\hat{\theta}$ is said to be asymptotically efficient if $\sqrt{n}\hat{\theta}$ has a variance that approaches the Cramér–Rao lower bound for large n; that is,

$$\lim_{n\to\infty} \mathrm{Var}(\sqrt{n}\hat{\theta}) = \frac{1}{-nE\left(\frac{\partial^2 \ln f(x;\theta)}{\partial\theta^2}\right)}. \tag{1.37}$$

1.3.1 Parameter Estimation

We now discuss some basic methods of parameter estimation, including the method of maximum likelihood estimation (MLE) and the method of moments. The assumption that the sample is representative of the population will be made both in the example and in later discussions.

Maximum Likelihood Estimation Method

In general, one deals with a sample density

$$f(x_1, x_2, \ldots, x_n) = f(x_1;\theta) f(x_2;\theta) \ldots f(x_n;\theta),$$

where $x_1, x_2, \ldots, x_n$ are random, independent observations of a population with density function $f(x)$.

For the general case, we would like to find an estimate or estimates, $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m$ (if such exist), where

$$f(x_1, x_2, \ldots, x_n; \theta_1, \theta_2, \ldots, \theta_m) >$$
$$f(x_1, x_2, \ldots, x_n; \theta_1', \theta_2', \ldots, \theta_m').$$

The notation $\theta_1', \theta_2', \ldots, \theta_n'$ refers to any other estimates different to $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m$.

Consider a random sample $X_1, X_2, \ldots, X_n$ from a distribution with a pdf $f(x;\theta)$. This distribution has a vector $\theta = (\theta_1, \theta_2, \ldots, \theta_m)'$ of unknown parameters associated with it, where m is the number of unknown parameters. Assuming that the random variables are independent, then the likelihood function, $L(X;\theta)$, is the product of the probability density function evaluated at each sample point

$$L(X,\theta) = \prod_{i=1}^{n} f(X_i;\theta), \tag{1.38}$$

where $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$. The maximum likelihood estimator $\hat{\theta}$ is found by maximizing $L(\boldsymbol{X};\theta)$ with respect to θ. In practice, it is often easier to maximize $\ln[L(\boldsymbol{X};\theta)]$ in order to find the vector of MLEs, which is valid because the logarithmic function is monotonic. The log likelihood function is given by

$$\ln L(X,\theta) = \sum_{i=1}^{n} \ln f(X_i;\theta) \tag{1.39}$$

and is asymptotically normally distributed since it consists of the sum of n independent variables and the central limit theorem is implied. Since $L(X;\theta)$ is a joint probability density function for $X_1, X_2, \ldots, X_n$, its integral must be 1; that is,

$$\int_0^\infty \int_0^\infty \cdots \int_0^\infty L(X;\theta)\,\mathrm{d}X = 1.$$

Assuming that the likelihood is continuous, the partial derivative of the left-hand side with respect to one of the parameters, θ_i, yields

$$\begin{aligned}
&\frac{\partial}{\partial\theta_i}\int_0^\infty \int_0^\infty \cdots \int_0^\infty L(X;\theta)\,\mathrm{d}X \\
&= \int_0^\infty \int_0^\infty \cdots \int_0^\infty \frac{\partial}{\partial\theta_i} L(X;\theta)\,\mathrm{d}X \\
&= \int_0^\infty \int_0^\infty \cdots \int_0^\infty \frac{\partial \log L\,(X;\theta)}{\partial\theta_i} L(X;\theta)\,\mathrm{d}X \\
&= E\left(\frac{\partial \log L\,(X;\theta)}{\partial\theta_i}\right) \\
&= E[U_i(\theta)] \quad \text{for } i = 1, 2, \ldots, m,
\end{aligned}$$

where $\boldsymbol{U}(\theta) = [U_1(\theta), U_2(\theta), \ldots U_n(\theta)]'$ is often called the score vector, and the vector $\boldsymbol{U}(\theta)$ has components

$$U_i(\theta) = \frac{\partial[\log L\,(X;\theta)]}{\partial\theta_i} \quad \text{for } i = 1, 2, \ldots, m \tag{1.40}$$

which, when equated to zero and solved, yields the MLE vector $\boldsymbol{\theta}$.

Suppose that we can obtain a nontrivial function of $X_1, X_2, \ldots, X_n$, say $h(X_1, X_2, \ldots, X_n)$, such that, when θ is replaced by $h(X_1, X_2, \ldots, X_n)$, the likelihood function L will achieve a maximum. In other words,

$$L[X, h(X)] \geq L(X,\theta)$$

for every θ. The statistic $h(X_1, X_2, \ldots, X_n)$ is called a maximum likelihood estimator of θ and will be denoted as

$$\hat{\theta} = h(x_1, x_2, \ldots, x_n). \tag{1.41}$$

The observed value of $\hat{\theta}$ is called the MLE of θ. In general, the mechanics for obtaining the MLE can be obtained as follows:

Step 1. Find the joint density function $L(X, \theta)$
Step 2. Take the natural log of the density $\ln L$
Step 3. Find the partial derivatives of $\ln L$ with respect to each parameter
Step 4. Set these partial derivatives to "zero"
Step 5. Solve for parameter(s).

Example 1.9: Let $X_1, X_2, \ldots, X_n$, denote a random sample from the normal distribution $N(\mu, \sigma^2)$. Then the likelihood function is given by

$$L(X, \mu, \sigma^2) = \left(\frac{1}{2\pi}\right)^{\frac{n}{2}} \frac{1}{\sigma^n} \mathrm{e}^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

and

$$\ln L = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2.$$

Thus, we have

$$\frac{\partial \ln L}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) = 0,$$
$$\frac{\partial \ln L}{\partial \sigma^2} = -\frac{n}{2\sigma^2} - \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i-\mu)^2 = 0.$$

Solving the two equations simultaneously, we obtain

$$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n},$$
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

Note that the MLEs, if they exist, are both sufficient and efficient estimates. They also have an additional property called invariance – in other words, for an MLE of θ, $\mu(\theta)$ is the MLE of $\mu(\theta)$. However, they are not necessarily unbiased (i. e., $E(\hat{\theta}) = \theta$). In fact, the point is that

$$E(\hat{\sigma}^2) = \left(\frac{n-1}{n}\right)\sigma^2 \neq \sigma^2.$$

Therefore, for small n, σ^2 is usually adjusted to account for this bias, and the best estimate of σ^2 is

$$\hat{\sigma}^2 = \left(\frac{1}{n-1}\right)\sum_{i=1}^{n}(x_i-\bar{x})^2.$$

Sometimes it is difficult, if not impossible, to obtain maximum likelihood estimators in a closed form, and therefore numerical methods must be used to maximize the likelihood function.

Example 1.10: Suppose that $X_1, X_2, \ldots, X_n$ is a random sample from the Weibull distribution with pdf

$$f(x, \alpha, \lambda) = \alpha\lambda x^{\alpha-1}\mathrm{e}^{-\lambda x^{\alpha}}. \tag{1.42}$$

The likelihood function is

$$L(X, \alpha, \lambda) = \alpha^n \lambda^n \prod_{i=1}^{n} x_i^{\alpha-1} \mathrm{e}^{-\lambda\sum_{i=1}^{n} x_i^{\alpha}}.$$

Then

$$\ln L = n\log\alpha + n\log\lambda + (\alpha-1)\sum_{i=1}^{n}\log x_i - \lambda\sum_{i=1}^{n} x_i^{\alpha},$$
$$\frac{\partial \ln L}{\partial \alpha} = \frac{n}{\alpha} + \sum_{i=1}^{n}\log x_i - \lambda\sum_{i=1}^{n} x_i^{\alpha}\log x_i = 0,$$
$$\frac{\partial \ln L}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i^{\alpha} = 0.$$

As noted, solutions of the above two equations for α and λ are extremely difficult to obtain and require the application of either graphical or numerical methods. It is sometimes desirable to use a quick method of estimation, which leads to a discussion of the method of moments.

Method of Moments

Here one simply sets the sample moments equal to the corresponding population moments. For example, for the gamma distribution, the mean and the variance of

the distribution are, respectively, $\frac{\alpha}{\beta}$ and $\frac{\alpha}{\beta^2}$. Therefore, one has the following two equations in two unknowns:

$$\bar{X} = \frac{\alpha}{\beta},$$
$$S^2 = \frac{\alpha}{\beta^2}.$$

Solving these two equations simultaneously, we obtain

$$\alpha = \frac{\bar{X}^2}{S^2},$$
$$\beta = \frac{\bar{X}}{S^2}.$$

1.3.2 Maximum Likelihood Estimation with Censored Data

Censored data arises when we monitor for a random variable of interest – unit failure, for example – but the monitoring is stopped before measurements are complete (i. e. before the unit fails). In other words, censored observation contains only partial information about the random variable of interest. In this section, we consider two types of censoring. The first type of censoring is called Type I censoring, where the event is only observed if it occurs prior to some prespecified time. The second type of censoring is Type II censoring, in which the study continues until the failure of the first r units (or components), where r is some predetermined integer ($r < n$).

Examples of Type II censoring are often used when testing equipment life. Here our items are tested at the same time, and the test is terminated when r of the n items have failed. These approaches may save time and resources because it may take a very long time for all of the items to fail. Both Type I and Type II censoring arise in many reliability applications.

For example, let's say that we have a batch of transistors or tubes. We begin to test them all at $t = 0$, and record their times to failure. Some transistors may take a long time to burn out, and we will not want to wait that long to end the experiment. We might stop the experiment at a prespecified time t_c, in which case we have Type I censoring. On the other hand, we might not know what fixed value to use for the censoring time beforehand, so we decide to wait until a prespecified number of units have failed, r, in which case we have Type II censoring.

Censoring times may vary from individual to individual or from application to application. We now discuss a general case known as multiple-censored data.

Parameter Estimate with Multiple-Censored Data

The likelihood function for multiple-censored data is given by

$$\begin{aligned} L &= f(t_{1,\mathrm{f}}, \ldots, t_{r,\mathrm{f}}, t_{1,\mathrm{s}}, \ldots, t_{m,\mathrm{s}}) \\ &= C \prod_{i=1}^{r} f(t_{i,\mathrm{f}}) \prod_{j=1}^{m} [1 - F(t_{j,\mathrm{s}})], \end{aligned} \tag{1.43}$$

where C is a constant, $f(.)$ is the density function and $F(.)$ is the distribution function. There are r failures at times $t_{1,\mathrm{f}}, \ldots, t_{r,\mathrm{f}}$ and m units with censoring times $t_{1,\mathrm{s}}, \ldots, t_{m,\mathrm{s}}$.

Note that we obtain Type-I censoring by simply setting $t_{i,\mathrm{f}} = t_{i,n}$ and $t_{j,\mathrm{s}} = t_0$ in the likelihood function in (1.43). The likelihood function for Type II censoring is similar to Type I censoring except $t_{j,\mathrm{s}} = t_r$ in (1.43). In other words, the likelihood function for the first r observations from a sample of size n drawn from the model in both Type I and Type II censoring is given by

$$L = f(t_{1,n}, \ldots, t_{r,n}) = C \prod_{i=1}^{r} f(t_{i,n}) [1 - F(t_*)]^{n-r}, \tag{1.44}$$

where $t_* = t_0$, the time of cessation of the test for Type I censoring and $t_* = t_r$, the time of the rth failure for Type II censoring.

Example 1.11: Consider a two-parameter probability density distribution with multiple-censored data and a distribution function with bathtub shaped failure rate, as given by [1.4]:

$$f(t) = \lambda\beta t^{\beta-1} \exp[t^{\beta} + \lambda(1 - \mathrm{e}^{t^{\beta}})], \; t, \lambda, \beta > 0 \tag{1.45}$$

and

$$F(t) = 1 - \exp[\lambda(1 - \mathrm{e}^{t^{\beta}})], \; t, \lambda, \beta > 0, \tag{1.46}$$

respectively.

Substituting the functions $f(t)$ and $F(t)$ into (1.45) and (1.46) into (1.44), we obtain the logarithm of the likelihood function:

$$\begin{aligned} \ln L = {} & \ln C + r \ln \lambda + r \ln \beta + \sum_{i=1}^{r} (\beta - 1) \ln t_i \\ & + (m + r)\lambda + \sum_{i=1}^{r} t_i^{\beta} - \left[\sum_{i=1}^{r} \lambda \mathrm{e}^{t_i^{\beta}} + \sum_{j=1}^{m} \lambda \mathrm{e}^{t_j^{\beta}} \right]. \end{aligned}$$

The function $\ln L$ can be maximized by setting the partial derivative of $\ln L$ with respect to λ and β equal to zero,

and solving the resulting equations simultaneously for λ and β. Therefore, we obtain

$$\frac{\partial \ln L}{\partial \lambda} = \frac{r}{\lambda} + (m+r) - \sum_{i=1}^{r} e^{t_i^{\beta}} - \sum_{j=1}^{m} e^{t_j^{\beta}} \equiv 0,$$

$$\frac{\partial \ln L}{\partial \beta} = \frac{r}{\beta} + \sum_{i=1}^{r} \ln t_i + \sum_{i=1}^{r} t_i^{\beta} \ln t_i - \lambda \left(\sum_{i=1}^{r} e^{t_i^{\beta}} t_i^{\beta} \ln t_i + \sum_{j=1}^{m} e^{t_j^{\beta}} t_j^{\beta} \ln t_j \right) \equiv 0.$$

This implies that

$$\hat{\lambda} = \frac{r}{\left(\sum_{i=1}^{r} e^{t_i^{\hat{\beta}}} + \sum_{j=1}^{m} e^{t_j^{\hat{\beta}}} \right) - m - r} \tag{1.47}$$

and that $\hat{\beta}$ is the solution of

$$\frac{r}{\hat{\beta}} + \sum_{i=1}^{r} \ln t_i + \sum_{i=1}^{r} t_i^{\hat{\beta}} \ln t_i = \frac{r}{\left(\sum_{i=1}^{r} e^{t_i^{\hat{\beta}}} + \sum_{j=1}^{m} e^{t_j^{\hat{\beta}}} \right) - m - r} \left(\sum_{i=1}^{r} e^{t_i^{\hat{\beta}}} t_i^{\hat{\beta}} \ln t_i + \sum_{j=1}^{m} e^{t_j^{\hat{\beta}}} t_j^{\hat{\beta}} \ln t_j \right). \tag{1.48}$$

We now discuss two special cases.

Case 1: Type I or Type II censored data

From (1.44), the likelihood function for the first r observations from a sample of size n drawn from the model in both Type I and Type II censoring is

$$L = f(t_{1,n}, \ldots, t_{r,n}) = C \prod_{i=1}^{r} f(t_{i,n}) [1 - F(t_*)]^{n-r},$$

where $t_* = t_0$, the test cessation time for Type I censoring, and $t_* = t_r$, the time of the rth failure for Type II censoring. Equations (1.47) and (1.48) become

$$\hat{\lambda} = \frac{r}{\sum_{i=1}^{r} e^{t_i^{\hat{\beta}}} + (n-r) e^{t_*^{\hat{\beta}}} - n},$$

$$\frac{r}{\hat{\beta}} + \sum_{i=1}^{r} \ln t_i + \sum_{i=1}^{r} t_i^{\hat{\beta}} \ln t_i = \frac{r}{\sum_{i=1}^{r} e^{t_i^{\hat{\beta}}} + (n-r) e^{t_*^{\hat{\beta}}} - n} \times \left(\sum_{i=1}^{r} e^{t_i^{\hat{\beta}}} t_i^{\hat{\beta}} \ln t_i + \sum_{j=1}^{m} e^{t_j^{\hat{\beta}}} t_j^{\hat{\beta}} \ln t_j \right)$$

Case 2: Complete censored data

Simply replace r with n in (1.47) and (1.48) and ignore the t_j portions. The maximum likelihood equations for the λ and β are given by

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} e^{t_i^{\hat{\beta}}} - n},$$

$$\frac{n}{\hat{\beta}} + \sum_{i=1}^{n} \ln t_i + \sum_{i=1}^{n} t_i^{\hat{\beta}} \ln t_i = \frac{n}{\sum_{i=1}^{n} e^{t_i^{\hat{\beta}}} - n} \times \sum_{i=1}^{n} e^{t_i^{\hat{\beta}}} t_i^{\hat{\beta}} \ln t_i .$$

Confidence Intervals of Estimates

The asymptotic variance–covariance matrix for the parameters (λ and β) is obtained by inverting the Fisher information matrix

$$I_{ij} = E\left(-\frac{\partial^2 L}{\partial \theta_i \partial \theta_j} \right), \quad i, j = 1, 2, \tag{1.49}$$

where $\theta_1, \theta_2 = \lambda$ or β [1.5]. This leads to

$$\begin{pmatrix} \mathrm{Var}(\hat{\lambda}) & \mathrm{Cov}(\hat{\lambda}, \hat{\beta}) \\ \mathrm{Cov}(\hat{\lambda}, \hat{\beta}) & \mathrm{Var}(\hat{\beta}) \end{pmatrix} = \begin{pmatrix} E\left(-\frac{\partial^2 \ln L}{\partial \lambda^2} |_{\hat{\lambda}, \hat{\beta}} \right) & E\left(-\frac{\partial^2 \ln L}{\partial \lambda \partial \beta} |_{\hat{\lambda}, \hat{\beta}} \right) \\ E\left(-\frac{\partial^2 \ln L}{\partial \beta \partial \lambda} |_{\hat{\lambda}, \hat{\beta}} \right) & E\left(-\frac{\partial^2 \ln L}{\partial^2 \beta} |_{\hat{\lambda}, \hat{\beta}} \right) \end{pmatrix}^{-1}. \tag{1.50}$$

We can obtain approximate $(1-\alpha)100\%$ confidence intervals for the parameters λ and β based on the asymptotic normality of the MLEs [1.5] as:

$$\hat{\lambda} \pm Z_{\alpha/2} \sqrt{\mathrm{Var}(\hat{\lambda})} \quad \text{and} \quad \hat{\beta} \pm Z_{\alpha/2} \sqrt{\mathrm{Var}(\hat{\beta})}, \tag{1.51}$$

where $Z_{\alpha/2}$ is the upper percentile of the standard normal distribution.

Application 1. Consider the lifetime of a part from a helicopter's main rotor blade. Data on lifetime of the part taken a system database collected from October 1995 to September 1999 [1.3] are shown in Table 1.2. In this application, we consider several distribution functions for this data, including Weibull, log normal, normal, and loglog distribution functions.

The Pham pdf with parameters a and α is

$$f(t) = \alpha(\ln a)t^{\alpha-1}a^{t^\alpha}\,\mathrm{e}^{1-a^{t^\alpha}} \quad \text{for } t>0,\ \alpha>0,\ a>1$$

and its corresponding log likelihood function (1.39) is

$$\begin{aligned}\log L(a,\alpha) &= n\log\alpha + n\ln(\ln a)\\ &\quad + (\alpha-1)\left(\sum_{i=1}^{n}\ln t_i\right)\\ &\quad + \ln a\cdot\sum_{i=1}^{n}t_i^{\alpha} + n - \sum_{i=1}^{n}a^{t_i^{\alpha}}\,.\end{aligned}$$

We then determine the confidence intervals for parameter estimates a and α. From the above log likelihood function, we can obtain the Fisher information matrix $\mathbf{H}$ as $\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22}\end{pmatrix}$, where

$$h_{11} = E\left(-\frac{\partial^2 \log L}{\partial a^2}\right),$$

$$h_{12} = h_{21} = E\left(-\frac{\partial^2 \log L}{\partial a\partial\alpha}\right),$$

$$h_{22} = E\left(-\frac{\partial^2 \log L}{\partial \alpha^2}\right).$$

The variance matrix $\mathbf{V}$ can be obtained as follows:

$$\mathbf{V} = (\mathbf{H})^{-1} = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22}\end{pmatrix}. \tag{1.52}$$

The variances of a and α are

$$\mathrm{Var}(a) = v_{11} \quad \mathrm{Var}(\alpha) = v_{22}.$$

Table 1.2 Main rotor blade data

Part code	Time to failure (h)	Part code	Time to failure (h)
xxx-015-001-107	1634.3	xxx-015-001-107	403.2
xxx-015-001-107	1100.5	xxx-015-001-107	2898.5
xxx-015-001-107	1100.5	xxx-015-001-107	2869.1
xxx-015-001-107	819.9	xxx-015-001-107	26.5
xxx-015-001-105	1398.3	xxx-015-001-107	26.5
xxx-015-001-107	1181	xxx-015-001-107	3180.6
xxx-015-001-107	128.7	xxx-015-001-107	644.1
xxx-015-001-107	1193.6	xxx-015-001-107	1898.5
xxx-015-001-107	254.1	xxx-015-001-107	3318.2
xxx-015-001-107	3078.5	xxx-015-001-107	1940.1
xxx-015-001-107	3078.5	xxx-015-001-107	3318.2
xxx-015-001-107	3078.5	xxx-015-001-107	2317.3
xxx-015-001-107	26.5	xxx-015-001-107	1081.3
xxx-015-001-107	26.5	xxx-015-001-107	1953.5
xxx-015-001-107	3265.9	xxx-015-001-107	2418.5
xxx-015-001-107	254.1	xxx-015-001-107	1485.1
xxx-015-001-107	2888.3	xxx-015-001-107	2663.7
xxx-015-001-107	2080.2	xxx-015-001-107	1778.3
xxx-015-001-107	2094.3	xxx-015-001-107	1778.3
xxx-015-001-107	2166.2	xxx-015-001-107	2943.6
xxx-015-001-107	2956.2	xxx-015-001-107	2260
xxx-015-001-107	795.5	xxx-015-001-107	2299.2
xxx-015-001-107	795.5	xxx-015-001-107	1655
xxx-015-001-107	204.5	xxx-015-001-107	1683.1
xxx-015-001-107	204.5	xxx-015-001-107	1683.1
xxx-015-001-107	1723.2	xxx-015-001-107	2751.4

One can approximately obtain the $(1-\beta)100\%$ confidence intervals for a and α based on the normal distribution as $\left[\hat{a}-z_{\frac{\beta}{2}}\sqrt{v_{11}}, \hat{a}+z_{\frac{\beta}{2}}\sqrt{v_{11}}\right]$ and $\left[\hat{\alpha}-z_{\frac{\beta}{2}}\sqrt{v_{22}}, \hat{\alpha}+z_{\frac{\beta}{2}}\sqrt{v_{22}}\right]$, respectively, where v_{ij} is given in (1.52) and z_β is $(1-\frac{\beta}{2})100\%$ of the standard normal distribution. Having obtained $\hat{a}$ and $\hat{\alpha}$, the MLE of the reliability function can be computed as

$$\hat{R}(t) = \mathrm{e}^{1-\hat{a}t^{\hat{\alpha}}}. \tag{1.53}$$

Let us define a partial derivative vector for reliability $R(t)$ as:

$$v[R(t)] = \left(\frac{\partial R(t)}{\partial a}\ \frac{\partial R(t)}{\partial \alpha}\right)$$

Then the variance of $R(t)$ can be obtained as

$$\mathrm{Var}\left[R(t)\right] = v[R(t)]\mathrm{V}\left(v[R(t)]\right)^{\mathrm{T}},$$

where V is given in (1.52).

One can approximately obtain the $(1-\beta)100\%$ confidence interval for $R(t)$ is

$$\left[\hat{R}(t) - z\beta\sqrt{\mathrm{Var}\left[R(t)\right]},\ \hat{R}(t) + z\beta\sqrt{\mathrm{Var}\left[R(t)\right]}\right].$$

The MLE parameter estimations for the loglog distribution and its corresponding parameters, based on the data set shown in Table 1.2, are:

$$\hat{\alpha} = 1.1075, \quad \mathrm{Var}(\hat{\alpha}) = 0.0162,$$
$$95\%\ \text{CI for}\ \hat{\alpha}: [0.8577, 1.3573];$$
$$\hat{a} = 1.0002, \quad \mathrm{Var}(\hat{a}) = 2.782\,\mathrm{e}^{-08},$$
$$95\%\ \text{CI for}\ a: [0.9998, 1.0005].$$

Similarly, the C.I. for $R(t)$ can be obtained directly using (1.53).

1.3.3 Statistical Change-Point Estimation Methods

The change-point problem has been widely studied in reliability applications in areas such as biological sciences, survival analysis and environmental statistics.

Assume that there is a sequence of random variables $X_1, X_2, \ldots, X_n$, that represents the inter-failure times, and that an index change-point τ exists, such that $X_1, X_2, \ldots, X_\tau$ have a common distribution F with a density function $f(t)$ and $X_{\tau+1}, X_{\tau+2}, \ldots, X_n$ have a distribution G with a density function $g(t)$, where $F \neq G$. Consider the following assumptions:

1. There is a finite but unknown number of units N to be tested.
2. At the beginning, all of the units have the same lifetime distribution F. After τ failures are observed, the remaining $(N-\tau)$ items have the distribution G. The change-point τ is assumed unknown.
3. The sequence $\{X_1, X_2, \ldots, X_\tau\}$ is statistically independent of the sequence $\{X_{\tau+1}, X_{\tau+2}, \ldots, X_n\}$.
4. The lifetime test is performed according to the Type II censoring approach, in which the number of failures n is predetermined.

Note that the total number of units to put up for testing N can be determined in advance in hardware reliability testing. However, in software reliability testing, the parameter N can be defined as the initial number of faults in the software, and it can be considered to be an unknown parameter. Let $T_1, T_2, \ldots, T_n$ be the arrival times for sequential failures. Then

$$\begin{aligned} T_1 &= X_1, \\ T_2 &= X_1 + X_2, \\ &\vdots \\ T_n &= X_1 + X_2 + \cdots X_n. \end{aligned} \tag{1.54}$$

The failure times $T_1, T_2, \ldots, T_\tau$ are the first τ order statistics of a sample of size N from the distribution F. The failure times $T_{\tau+1}, T_{\tau+2}, \ldots, T_n$ are the first $(n-\tau)$ order statistics of a sample of size $(N-\tau)$ from the distribution G.

Example 1.12: The Weibull change-point model of the lifetime distributions F and G with parameters (λ_1, β_1) and (λ_2, β_2), respectively, can be expressed as

$$F(t) = 1 - \exp\left(-\lambda_1 t^{\beta_1}\right), \tag{1.55}$$

$$G(t) = 1 - \exp\left(-\lambda_2 t^{\beta_2}\right). \tag{1.56}$$

Assume that the distributions belong to parametric families $\{F(t \mid \theta_1), \theta_1 \in \Theta_1\}$ and $\{G(t \mid \theta_2), \theta_2 \in \Theta_2\}$. Assume that $T_1, T_2, \ldots, T_\tau$ are the first τ order statistics of a sample of size N from the distribution $\{F(t \mid \theta_1), \theta_1 \in \Theta_1\}$ and that $T_{\tau+1}, T_{\tau+2}, \ldots, T_n$ are the first $(n-\tau)$ order statistics of a sample of size $(N-\tau)$ from the distribution $\{G(t \mid \theta_2), \theta_2 \in \Theta_2\}$, where N is unknown. The log likelihood function can be expressed

as follows [1.6]:

$$L(\tau, N, \theta_1, \theta_2 \mid T_1, T_2, \ldots, T_n) = \sum_{i=1}^{n}(N-i+1) + \sum_{i=1}^{\tau} f(T_i \mid \theta_1) + \sum_{i=\tau+1}^{n} g(T_i \mid \theta_2) + (N-\tau)\log[1-F(T_\tau \mid \theta_1)] + (N-n)\log[1-G(T_n \mid \theta_2)]. \quad (1.57)$$

If the parameter N is known in which where hardware reliability is commonly considered for example, then the likelihood function is given by

$$L(\tau, \theta_1, \theta_2 \mid T_1, T_2, \ldots, T_n) = \sum_{i=1}^{\tau} f(T_i \mid \theta_1) + \sum_{i=\tau+1}^{n} g(T_i \mid \theta_2) + (N-\tau)\log[1-F(T_\tau \mid \theta_1)] + (N-n)\log[1-G(T_n \mid \theta_2)].$$

The maximum likelihood estimator (MLE) of the change-point value $\hat{\tau}$ and $(\hat{N}, \hat{\theta}_1, \hat{\theta}_2)$ can be obtained by taking partial derivatives of the log likelihood function in (1.57) with respect to the unknown parameters that maximize the function. It should be noted that there is no closed form for $\hat{\tau}$, but it can be obtained by calculating the log likelihood for each possible value of τ, $1 \le \tau \le (n-1)$, and selecting the value that maximizes the log likelihood function.

Application 2: A Software Model with a Change Point

In this application, we examine the case where the sample size N is unknown. Consider a software reliability model developed by *Jelinski* and *Moranda* in 1972, often called the Jelinski–Moranda model. The assumptions of the model are as follows:

1. There are N initial faults in the program.
2. A detected fault is removed instantaneously and no new fault is introduced.
3. Each failure caused by a fault occurs independently and randomly in time according to an exponential distribution.
4. The functions F and G are exponential distributions with failure rate parameters λ_1 and λ_2, respectively.

Based on these assumptions, the inter-failure times $X_1, X_2, \ldots, X_n$ are independently exponentially distributed. Specifically, $X_i = T_i - T_{i-1}$, $i = 1, 2, \ldots \tau$, are exponentially distributed with parameter $\lambda_1 (N-i+1)$, where λ_1 is the initial fault detection rate of the first τ failures, and $X_j = T_j - T_{j-1}$, $j = \tau+1, \tau+2, \ldots n$ are exponentially distributed with parameter $\lambda_2 (N-\tau-j+1)$, where λ_2 is the fault detection rate of the first $n-\tau$ failures. If $\lambda_1 = \lambda_2$, it means that each fault removal is the same and that the change-point model becomes the Jelinski–Moranda software reliability model [1.7].

The MLEs of the parameters $(\tau, N, \lambda_1, \lambda_2)$ can be obtained by solving the following equations simultaneously:

$$\hat{\lambda}_1 = \frac{\tau}{\sum_{i=1}^{\tau}(\hat{N}-i+1)x_i}, \quad (1.58)$$

$$\hat{\lambda}_2 = \frac{(n-\tau)}{\sum_{i=\tau+1}^{n}(\hat{N}-i+1)x_i}, \quad (1.59)$$

$$\sum_{i=1}^{n}\frac{1}{(\hat{N}-i+1)} = \hat{\lambda}_1\sum_{i=1}^{\tau}x_i + \hat{\lambda}_2\sum_{i=\tau+1}^{n}x_i. \quad (1.60)$$

To illustrate the model, we use the data set shown in Table 1.3 to obtain the unknown parameters $(\tau, N, \lambda_1, \lambda_2)$ using (1.58)–(1.60). The data in Table 1.3 [1.8] shows the successive inter-failure times for a real-time command and control system. The table reads from left to right in rows, and the recorded times are execution times, in seconds. There are 136 failures in total. Figure 1.6 plots the log-likelihood function versus the number of failures. The MLEs of the parameters $(\tau, N, \lambda_1, \lambda_2)$ with one change point are given by

$$\hat{\tau} = 16,\ \hat{N} = 145,\ \hat{\lambda}_1 = 1.1\times10^{-4},\ \hat{\lambda}_2 = 0.31\times10^{-4}.$$

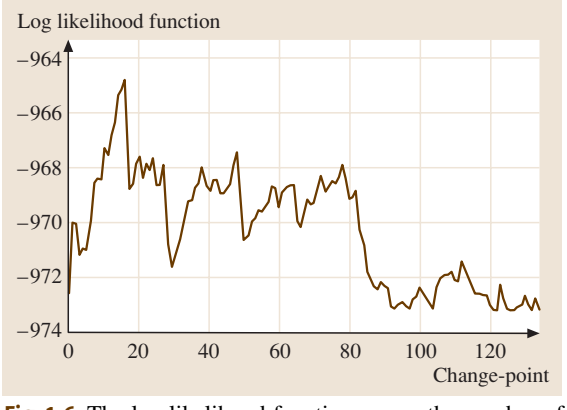


Fig. 1.6 The log likelihood function versus the number of failures

Table 1.3 Successive inter-failure times (in s) for a real-time command system

3	30	113	81	115	9	2	91	112	15
138	50	77	24	108	88	670	120	26	114
325	55	242	68	422	180	10	1146	600	15
36	4	0	8	227	65	176	58	457	300
97	263	452	255	197	193	6	79	816	1351
148	21	233	134	357	193	236	31	369	748
0	232	330	365	1222	543	10	16	529	379
44	129	810	290	300	529	281	160	828	1011
445	296	1755	1064	1783	860	983	707	33	868
724	2323	2930	1461	843	12	261	1800	865	1435
30	143	108	0	3110	1247	943	700	875	245
729	1897	447	386	446	122	990	948	1082	22
75	482	5509	100	10	1071	371	790	6150	3321
1045		648	5485	1160	1864	4116			

If we do not consider a change point in the model, the MLEs of the parameters N and λ, can be given as

$$\hat{N} = 142,\ \hat{\lambda} = 0.35 \times 10^{-4}.$$

From Fig. 1.6,

it is clear that it is worth considering change points in reliability functions.

1.3.4 Goodness of Fit Techniques

The problem discussed here is one of comparing an observed sample distribution with a theoretical distribution. Two common techniques that will be discussed are the χ^2 goodness-of-fit test and the Kolmogorov–Smirnov "d" test.

Chi-Squared Test

The following statistic

$$\chi^2 = \sum_{i=1}^{k} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \tag{1.61}$$

has a chi-squared (χ^2) distribution with k degrees of freedom. The procedure used for the chi-squared test is:

1. Divide the sample data into mutually exclusive cells (normally 8–12) such that the range of the random variable is covered.
2. Determine the frequency, f_i, of sample observations in each cell.
3. Determine the theoretical frequency, F_i, for each cell (the area under density function between cell boundaries X_n – total sample size). Note that the theoretical frequency for each cell should be greater than 1. This step normally requires estimates for the population parameters, which can be obtained from the sample data.
4. Form the statistic

$$A = \sum_{i=1}^{k} \frac{(f_i - F_i)^2}{F_i}. \tag{1.62}$$

5. From the χ^2 tables, choose a value of χ^2 with the desired significance level and degrees of freedom ($= k - 1 - r$, where r is the number of population parameters estimated).
6. Reject the hypothesis that the sample distribution is the same as the theoretical distribution if

$$A > \chi^2_{1-\alpha,k-1-r},$$

where α is called the significance level.

Example 1.13: Given the data in Table 1.4, can the data be represented by the exponential distribution with a significance level of α?

From the above calculation, $\hat{\lambda} = 0.002\,63$, $R_i = \mathrm{e}^{-\lambda t_i}$ and $Q_i = 1 - R_i$. Given that the significance level α is 0.1, from (1.62), we obtain

$$A = \sum_{i=1}^{11} \frac{(f_i - F_i)^2}{F_i} = 6.165\,.$$

From Table 1.9 in Sect. 1.A, the value of χ^2 with nine degrees of freedom and $\alpha = 0.1$ is 14.68; that is,

$$\chi^2_{9,0.1} = 14.68\,.$$

Since $S = 6.165 < 14.68$, we would not reject the hypothesis of an exponential with $\lambda = 0.002\,63$.

Table 1.4 Sample observations for each cell boundary

Cell boundaries	f_i	$Q_i = (1 - R_i)\,60$	$F_i = Q_i - Q_{i-1}$
0 – 100	10	13.86	13.86
100 – 200	9	24.52	10.66
200 – 300	8	32.71	8.19
300 – 400	8	39.01	6.30
400 – 500	7	43.86	4.85
500 – 600	6	47.59	3.73
600 – 700	4	50.45	2.86
700 – 800	4	52.66	2.21
800 – 900	2	54.35	1.69
900 – 1000	1	55.66	1.31
> 1000	1	58.83	2.17

If in the statistic

$$A = \sum_{i=1}^{k} \left(\frac{f_i - F_i}{\sqrt{F_i}} \right)^2, \quad \left(\frac{f_i - F_i}{\sqrt{F_i}} \right)$$

is approximately normal for large samples, then A also has a χ^2 distribution. This is the basis for the goodness of fit test.

Kolmogorov–Smirnov d Test

Both the χ^2 and "d" tests are nonparametric tests. However, the χ^2 test largely assumes sample normality of the observed frequency about its mean, while "d" assumes only a continuous distribution. Let $X1 \le X_2 \le X_3 \le \ldots \le X_n$ denote the ordered sample values. Define the observed distribution function, $F_n(x)$, as:

$$F_n(X) = \begin{cases} 0 & \text{for } x \le x_1 \\ \frac{i}{n} & \text{for } x_i < x \le x_{i+1} \\ 1 & \text{for } x > x_n \end{cases} .$$

Assume the testing hypothesis

$$H_0 : F(x) = F_0(x),$$

where $F_0(x)$ is a given continuous distribution and $F(x)$ is an unknown distribution. Let

$$d_n = \sup_{-\infty < x < \infty} |F_n(x) - F_0(x)|.$$

Since $F_0(x)$ is a continuous increasing function, we can evaluate $|F_n(x) - F_0(x)|$ for each n. If $d_n \le d_{n,\alpha}$, then we will not reject the hypothesis H_0; otherwise, we will reject it when $d_n > d_{n,\alpha}$. The value of $d_{n,\alpha}$ can be found in Table 1.10 in Sect. 1.A, where n is the sample size and a is the level of significance.

1.3.5 Least Squared Estimation

One common approach to curve fitting, which is unrelated to normal regression theory and MLE estimates of coefficients but uses identical formulae, is called the method of least squares. This method is based on minimizing the sum of the squared distances from the best fit line to the actual data points. It just so happens that finding the MLEs for the coefficients of the regression line also involves this sum of squared distances.

Normal Linear Regression

Regression considers the distribution of one variable as a function of another when the other variable is fixed at each of several levels. In the normal bivariate case, consider the distribution of X as a function of given values of Z where $X = \alpha + \beta Z$. Consider a sample of n observations (x_i, z_i). We can obtain the likelihood and its natural log for the normal distribution as follows:

$$\begin{aligned} & f(x_1, x_2, \ldots, x_n) \\ & \quad = \frac{1}{2\pi^{\frac{n}{2}}} \left(\frac{1}{\sigma^2} \right)^{\frac{n}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \alpha - \beta z_i)^2}, \\ & \ln L = -\frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 \\ & \quad - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \alpha - \beta z_i)^2. \end{aligned}$$

Taking the partial derivatives of $\ln L$ with respect to α and β, we have

$$\begin{aligned} \frac{\partial \ln L}{\partial \alpha} &= \sum_{i=1}^{n} (x_i - \alpha - \beta z_i)^2 = 0, \\ \frac{\partial \ln L}{\partial \beta} &= \sum_{i=1}^{n} z_i (x_i - \alpha - \beta z_i) = 0. \end{aligned}$$

The solutions to the simultaneous equations are

$$\begin{aligned} \hat{\alpha} &= \bar{X} - \beta \bar{Z}, \\ \hat{\beta} &= \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Z_i - \bar{Z})}{\sum_{i=1}^{n} (Z_i - \bar{Z})^2}. \end{aligned} \tag{1.63}$$

Least Squared Straight Line Fit

Assume that there is a linear relationship between X and $E(Y|x)$; that is, that $E(Y|x) = a + bx$. Given a set of data, we want to estimate the coefficients a and b

that minimize the sum of the squares. Suppose that the desired polynomial, $p(x)$, is written as

$$\sum_{i=0}^{m} a_i x^i,$$

where $a_0, a_1, \ldots, a_m$ are to be determined. The method of least squares chooses as "solutions" those coefficients that minimize the sum of the squares of the vertical (y) distances from the data points to the presumed polynomial. This means that the "best" polynomial is the one whose coefficients minimize the function L, where

$$L = \sum_{i=1}^{n} [y_i - p(x_i)]^2.$$

Here, we will only treat the linear case, where $X = \alpha + \beta Z$. The procedure for higher order polynomials is identical, although the computations become much more tedious. Assume a straight line of the form $X = \alpha + \beta Z$. For each observation (x_i, z_i) : $X_i = \alpha + \beta Z_i$, let

$$Q = \sum_{i=1}^{n} (x_i - \alpha - \beta z_i)^2.$$

We wish to find estimates for α and β that minimize Q. Taking partial differentials, we obtain

$$\frac{\partial Q}{\partial \alpha} = -2 \sum_{i=1}^{n} (x_i - \alpha - \beta z_i) = 0,$$

$$\frac{\partial Q}{\partial \beta} = -2 \sum_{i=1}^{n} z_i (x_i - \alpha - \beta z_i) = 0.$$

Note that the above are the same as the MLE equations for normal linear regression. Therefore, we obtain the following results:

$$\hat{\alpha} = \bar{x} - \beta \bar{z},$$

$$\hat{\beta} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(z_i - \bar{z})}{\sum_{i=1}^{n} (z_i - \bar{z})^2}. \tag{1.64}$$

The above gives an example of least squares applied to a linear case. The same pattern applies for higher-order curves with 3, 4 and so on solutions.

1.3.6 Interval Estimation

A point estimate is sometimes inadequate at providing an estimate for an unknown parameter, since it rarely coincides with the true value of the parameter. An alternative way is to obtain a confidence interval estimation of the form $[\theta_L, \theta_U]$ where θ_L is the lower bound and θ_U is the upper bound.

Point estimates can become more useful if some measure of their error is given; in other words, if some kind of tolerance for their high and low values is given. Thus, if an interval estimator is $[\theta_L, \theta_U]$ with a given probability $(1-\alpha)$, then θ_L and θ_U are called the $100\,(1-\alpha)\,\%$ confidence limits for the given parameter θ, and the interval between them is a $100\,(1-\alpha)\,\%$ confidence interval, while $(1-\alpha)$ is called the confidence coefficient.

Confidence Intervals for Normal Parameters

The one-dimensional normal distribution has two parameters: mean μ and variance σ^2. The simultaneous employment of both parameters in a confidence statement concerning percentages of the population will be discussed in the next section on tolerance limits. Hence, individual confidence statements about μ and σ^2 will be discussed here.

Confidence Limits for a Mean μ with a Known σ^2. It is easy to show that the statistic

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

is a standard normal distribution, where

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Hence, a $100\,(1-\alpha)\,\%$ confidence interval for the mean μ is given by

$$P\left[\bar{X} - Z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + Z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}\right] = 1 - \alpha. \tag{1.65}$$

In other words,

$$\mu_L = \bar{X} - Z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \quad \text{and} \quad \mu_U = \bar{X} + Z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}.$$

Example 1.14: Draw a sample of size 4 from a normal distribution with a known variance = 9, say $x_1 = 2$, $x_2 = 3$, $x_3 = 5$, $x_4 = 2$. Determine the location of the true mean (μ). The sample mean can be calculated as

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{2+3+5+2}{4} = 3.$$

Assuming that $\alpha = 0.05$ and, from the standard normal distribution (Table 1.6 in Sect. 1.A), $Z_{0.025} = 1.96$, then we obtain

$$P\left[3 - 1.96\frac{3}{\sqrt{4}} < \mu < 3 + 1.96\frac{3}{\sqrt{4}}\right] = 0.95,$$
$$P[0.06 < \mu < 5.94] = 0.95.$$

This example shows that there is a 95% probability that the true mean is somewhere between 0.06 and 5.94. Now, μ is a fixed parameter and does not vary, so how do we interpret the probability? If samples of size 4 were to be repeatedly drawn, a different set of limits would be constructed each time. If this is the case, the interval becomes the random variable and the interpretation is that, for 95% of the time, the interval constructed in this way will contain the true (fixed) parameter.

Confidence Limits for a Mean μ with an Unknown σ^2. Let

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}. \tag{1.66}$$

It can be shown that the statistic

$$T = \frac{\bar{X} - \mu}{\frac{S}{\sqrt{n}}}$$

has a t distribution with $(n-1)$ degrees of freedom (see Table 1.7 in Appendix A). Thus, for a given sample mean and sample standard deviation, we obtain

$$P\left[|T| < t_{\frac{\alpha}{2},n-1}\right] = 1 - \alpha.$$

Hence, a $100\,(1-\alpha)\,\%$ confidence interval for the mean μ is given by

$$p\left[\bar{X} - t_{\frac{\alpha}{2},n-1}\frac{S}{\sqrt{n}} < \mu < \bar{X} + t_{\frac{\alpha}{2},n-1}\frac{S}{\sqrt{n}}\right]$$
$$= 1 - \alpha. \tag{1.67}$$

Example 1.15: The variability of a new product was investigated. An experiment was run using a sample of size $n = 25$; the sample mean was found to be $\bar{X} = 50$ and the variance $\sigma^2 = 16$. From Table 1.7 in Appendix A, $t_{\frac{\alpha}{2},n-1} = t_{0.025,24} = 2.064$. The 95% confidence limit for μ is given by

$$P\left[50 - 2.064\sqrt{\frac{16}{25}} < \mu\right.$$
$$\left. < 50 + 2.064\sqrt{\frac{16}{25}}\right] = 0.95\,,$$
$$P[48.349 < \mu < 51.651]x = 0.95\,.$$

Note that, for one-sided limits, one should choose t_α, or $t_{1-\alpha}$.

Confidence Limits on σ^2. Note that $n\hat{\sigma}^2/\sigma^2$ has a χ^2 distribution with $(n-1)$ degrees of freedom. Correcting for the bias in $\hat{\sigma}^2$, it is clear that $(n-1)\hat{\sigma}^2/\sigma^2$ has this same distribution. Hence,

$$P\left[\chi^2_{\frac{\alpha}{2},n-1} < \frac{(n-1)S^2}{\sigma^2} < \chi^2_{1-\frac{\alpha}{2},n-1}\right] = 1 - \alpha$$

or

$$P\left[\frac{\sum(x_i - \bar{x})^2}{\chi^2_{1-\frac{\alpha}{2},n-1}} < \sigma^2 < \frac{\sum(x_i - \bar{x})^2}{\chi^2_{\frac{\alpha}{2},n-1}}\right] = 1 - \alpha. \tag{1.68}$$

Again, for one-sided limits, one should choose $\chi^2(\alpha)$ or $\chi^2(1-\alpha)$.

Confidence Intervals for Exponential Parameters

The pdf and cdf for the exponential distribution are

$$f(x) = \lambda\,\mathrm{e}^{-\lambda x}, \quad x > 0, \lambda > 0$$

and

$$F(x) = 1 - \mathrm{e}^{-\lambda x},$$

respectively. It was shown that the distribution of a function of the estimate

$$\hat{\lambda} = \frac{r}{\sum_{i=1}^{n} x_i + (n-r)x_r} \tag{1.69}$$

derived from a test of n identical components with common exponential failure density (failure rate λ), whose testing was stopped after the rth failure, was chi-squared (χ^2), i. e.,

$$2r\frac{\lambda}{\hat{\lambda}} = 2\lambda T$$
(χ^2 distribution with $2r$ degrees of freedom),

where T is the total time accrued by all units. Knowing the distribution of $2\lambda T$ allows us to obtain the confidence limits on the parameter as:

$$P\left[\chi^2_{1-\frac{\alpha}{2},2r} < 2\lambda T < \chi^2_{\frac{\alpha}{2},2r}\right] = 1 - \alpha$$

or, equivalently, that

$$P\left[\frac{\chi^2_{1-\frac{\alpha}{2},2r}}{2T} < \lambda < \frac{\chi^2_{\frac{\alpha}{2},2r}}{2T}\right] = 1 - \alpha.$$

Table 1.5 Confidence limits for θ

Confidence limits	Fixed number of failures	Fixed time
One-sided lower limit	$\frac{2T}{\chi^2_{\alpha,2r}}$	$\frac{2T}{\chi^2_{\alpha,2r+2}}$
One-sided upper limit	$\frac{2T}{\chi^2_{1-\alpha,2r}}$	$\frac{2T}{\chi^2_{1-\alpha,2r}}$
Two-sided limits	$\frac{2T}{\chi^2_{\alpha/2,2r}}, \frac{2T}{\chi^2_{1-\alpha/2,2r}}$	$\frac{2T}{\chi^2_{\alpha/2,2r+2}}, \frac{2T}{\chi^2_{1-\alpha/2,2r}}$

This means that in $(1-\alpha)\%$ of the samples of a given size n, the random interval

$$\left(\frac{\chi^2_{1-\frac{\alpha}{2},2r}}{2T}, \frac{\chi^2_{\frac{\alpha}{2},2r}}{2T}\right)$$

will contain a population of constant failure rate. For $\theta = 1/\lambda$ or the MTBF, the above confidence limits change to

$$P\left[\frac{2T}{\chi^2_{\frac{\alpha}{2},2r}} < \theta < \frac{2T}{\chi^2_{1-\frac{\alpha}{2},2r}}\right] = 1-\alpha.$$

If testing is stopped at a fixed time rather than a fixed number of failures, the number of degrees of freedom in the lower limit increases by two. Table 1.5 shows the confidence limits for θ, the mean of the exponential density.

Confidence Intervals for Binomial Parameters

Consider a sequence of n Bernoulli trials with k successes and $(n-k)$ failures. We now determine one-sided upper and lower and two-sided limits on the parameter p, the probability of success. For the lower limit, the binomial sum is set up such that the probability of k or more successes with a true p as low as p_L is only $\alpha/2$. This means that the probability of k or more successes with a true p higher than p_L is $\left(1-\frac{\alpha}{2}\right)$.

$$\sum_{i=k}^{n}\binom{n}{i}p_L^i(1-p_L)^{n-i} = \frac{\alpha}{2}.$$

Similarly, the binomial sum for the upper limit is

$$\sum_{i=k}^{n}\binom{n}{i}p_U^i(1-p_U)^{n-i} = 1-\frac{\alpha}{2}$$

or, equivalently,

$$\sum_{i=0}^{k-1}\binom{n}{i}p_U^i(1-p_U)^{n-i} = \frac{\alpha}{2}.$$

Solving for p_L and p_U in the above equations,

$$P[p_L < p < p_U] = 1-\alpha.$$

For one-sided limits, merely change $\alpha/2$ to α.

Example 1.16: Given $n = 100$ with 25 successes, and 75 failures, an 80% two-sided confidence limit on p can be obtained as follows:

$$\sum_{i=25}^{100}\binom{100}{i}p_L^i(1-p_L)^{100-i} = 0.10,$$

$$\sum_{i=0}^{24}\binom{100}{i}p_U^i(1-p_U)^{100-i} = 0.10.$$

Solving the above two equations simultaneously, we obtain

$$p_L \approx 0.194 \quad \text{and} \quad p_U \approx 0.313,$$
$$P[0.194 < p < 0.313] = 0.80.$$

Continuing with Example 1.16 above, we now find an 80% one-sided confidence limit on p.

We start by setting the top equation to 0.20 and solving for p_L. It is then easy to obtain $p_L = 0.211$ and $P[p > 0.211] = 0.80$.

Let us now define $\bar{p} = k/n$, the number of successes divided by the number of trials. For large values of n and if $np > 5$ and $n(1-p) > 5$, and from the central limit theorem [1.9], the statistic

$$Z = \frac{(\bar{p}-p)}{\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}}$$

approximates to the standard normal distribution. Hence

$$P[-z_{\frac{\alpha}{2}} < Z < z_{\frac{\alpha}{2}}] = 1-\alpha.$$

Then

$$P\left[\bar{p} - z_{\frac{\alpha}{2}}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}} < p < \bar{p} + z_{\frac{\alpha}{2}}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}\right] = 1-\alpha.$$

Example 1.17: Find the two-sided confidence limit for $n = 900$, $k = 180$, and $\alpha = 0.05$. Then we obtain $p = 180/900 = 0.2$ and

$$P\left[0.2 - 1.96\sqrt{\frac{0.2(0.8)}{900}} < p < 0.2 + 1.96\sqrt{\frac{0.2(0.8)}{900}}\right] = 0.95,$$

$$P[0.174 < p < 0.226] = 0.95.$$

Confidence Intervals for Poisson Parameters

Limits for the Poisson parameters are completely analogous to those for the binomial distribution except that the sample space is denumerable instead of finite. The lower and upper limits can be solved simultaneously in the following equations:

$$\sum_{i=k}^{\infty} \frac{\lambda_{\mathrm{L}}^{i} \mathrm{e}^{-\lambda_{\mathrm{L}}}}{i!} = \frac{\alpha}{2},$$

$$\sum_{i=k}^{\infty} \frac{\lambda_{\mathrm{U}}^{i} \mathrm{e}^{-\lambda_{\mathrm{U}}}}{i!} = 1 - \frac{\alpha}{2},$$

or, equivalently,

$$\sum_{i=k}^{\infty} \frac{\lambda_{\mathrm{L}}^{i} \mathrm{e}^{-\lambda_{\mathrm{L}}}}{i!} = \frac{\alpha}{2},$$

$$\sum_{i=0}^{k-1} \frac{\lambda_{\mathrm{U}}^{i} \mathrm{e}^{-\lambda_{\mathrm{U}}}}{i!} = \frac{\alpha}{2}.$$

The one-sided limits are constructed in the same way as for binomial limits.

1.3.7 Nonparametric Tolerance Limits

Nonparametric tolerance limits are based on the smallest and largest observation in the sample, designated X_{S} and X_{L}, respectively. Due to their nonparametric nature, these limits are quite insensitive, and obtaining precisions similar to the parametric methods necessitates much larger samples. An interesting question here is to determine the sample size required to include at least $100\,(1-\alpha)\,\%$ of the population between X_{S} and X_{L} with a given probability y.

For two-sided tolerance limits, if $(1-\alpha)$ is the minimum proportion of the population contained between the largest observation X_{L} and the smallest observation X_{S} with a confidence $(1-\gamma)$, then it can be shown that

$$n(1-\alpha)^{n-1} - (n-1)(1-\alpha)^{n} = \gamma.$$

Therefore, the number of observations required is given by

$$n = \left(\frac{(2-\alpha)}{4\alpha} \chi^2_{1-\gamma,4} + \frac{1}{2} \right) + 1,$$

where a value of $\chi^2_{1-\gamma,4}$ is given in Table 1.8 of Sect. 1.A.

Example 1.18: Determine the tolerance limits that include at least 90% of the population with probability 0.95. Here,

$$\alpha = 0.1,\ \gamma = 0.95 \quad \text{and} \quad \chi^2_{0.05,4} = 9.488.$$

Therefore, a sample of size

$$n = \left[\frac{(2-0.1)}{4(0.1)} (9.488) + \frac{1}{2} \right] + 1 = 46$$

is required. For a one-sided tolerance limit, the number of observations required is given by

$$n = \left(\frac{\log(1-\gamma)}{\log(1-\alpha)} \right) + 1.$$

Example 1.19: As in Example 1.18, we wish to find a lower tolerance limit; that is, the number of observations required such that the probability is 0.95 that at least 90% of the population will exceed X_{S}. This is given by

$$n = \left(\frac{\log(1-0.95)}{\log(1-0.1)} \right) + 1 = 30.$$

One can easily generate a table containing the sample size required to include a given percentage of the population between X_{S} and X_{L} with a given confidence, or the sample size required to include a given percentage of the population above or below X_{S} or X_{L}, respectively.

1.3.8 Sequential Sampling

A sequential sampling scheme is one in which items are drawn one at a time and the results at any stage determine whether sampling or testing should stop. Thus, any sampling procedure for which the number of observations is a random variable can be regarded as sequential sampling. Sequential tests derive their name from the fact that the sample size is not determined in advance, but allowed to "float" with a decision (accept, reject, or continue test) after each trial or data point.

In general, let us consider the hypothesis

$$H_0 : f(x) = f_0(x) \text{ versus } H_1 : f(x) = f_1(x).$$

For an observational test, say X_1, if $X_1 \le A$, then we will accept the testing hypothesis $[H_0 : f(x) = f_0(x)]$; if $X_1 \ge A$ we will reject H_0 and accept $H_I : f(x) = f_1(x)$. Otherwise, we will continue to perform at least one more test. The interval $X_1 \le A$ is called the acceptance region. The interval $X_1 \ge A$ is called the rejection or critical region.

A "good" test is one that makes the α and β errors as small as possible. However, there is not much freedom to do this without increasing the sample size. A common procedure is to fix the β error and then choose a critical region to minimize the error or to maximize the "power"

(power $= 1-\beta$) of the test, or to choose the critical region so as to equalize the α and β errors to reasonable levels.

One criterion (similar to the MLE) used to construct tests is called the "probability ratio", which is the ratio of the sample densities under H_1/H_0. Consider the ratio of probabilities

$$\lambda_m = \frac{\prod_{i=1}^{n} f_1(x_i)}{\prod_{i=1}^{n} f_0(x_i)} > k.$$

Here, $x_1, x_2, \ldots, x_n$ are n independent random observations and k is chosen to give the desired a error.

Recall from the MLE discussion in Sect. 1.3.1 that $f_1(x_1), f_1(x_2), \ldots, f_1(x_n)$ are maximized under H_1 when the parameter(s), e.g. $\theta = \theta_1$ and similarly $f_0(x_1), f_0(x_2), \ldots, f_0(x_n)$, are maximized when $\theta = \theta_0$. Thus, the ratio will become large if the the sample favors H_1 and will become small if the sample favors H_0. Therefore, the test will be called a sequential probability ratio test if we

1. stop sampling and reject H_0 as soon as $\lambda_m \geq A$;
2. stop sampling and accept H_0 as soon as $\lambda_m \leq B$;
3. continue sampling as long as $B < \lambda_m < A$, where $A > B$.

The selection of A and B using the above test, as suggested by *Wald* (see [1.9]), can be determined as follows:

$$B = \frac{\beta}{1-\alpha} \quad \text{and} \quad A = \frac{1-\beta}{\alpha}$$

The bases for α and β are therefore:

$$P[\lambda_m > A|H_0] = \alpha$$
$$P[\lambda_m < A|H_1] = \beta$$

1.3.9 Bayesian Methods

The Bayesian approach to statistical inference is based on a theorem first presented by Thomas Bayes. To demonstrate the approach, let X have a pdf $f(x)$, which is dependent on θ. In the traditional statistical inference approach, θ is an unknown parameter, and hence is a constant. We now describe our prior supposition for the value of θ by a pdf of $h(\theta)$. This amounts to quantitatively assessing subjective judgment and should not be confused with the so-called objective probability assessment derived from the long-term frequency approach. Thus, θ will now essentially be treated as a random variable θ with a pdf of $h(\theta)$.

Consider a random sample $X_1, X_2, \ldots, X_n$ from $f(x)$ and define a statistic Y as a function of this random sample. Then there exists a conditional pdf $g(y|\theta)$ of Y for a given θ. The joint pdf for y and θ is

$$f(\theta, y) = h(\theta)g(y|\theta).$$

If θ is continuous, then

$$f_1(y) = \int_{\theta} h(\theta)g(y|\theta)\,\mathrm{d}\theta$$

is the marginal pdf for the statistic y. Given the information y, the conditional pdf for θ is

$$k(\theta|y) = \frac{h(\theta)g(y|\theta)}{f_1(y)} \quad \text{for } f_1(y) > 0$$
$$= \frac{h(\theta)g(y|\theta)}{\int_{\theta} h(\theta)g(y|\theta)\,\mathrm{d}\theta}$$

If θ is discrete, then

$$f_1(y) = \sum_k P(\theta_k)P(y|\theta_k)$$

and

$$P(\theta_i|y_i) = \frac{P(\theta_k)P(y_i|\theta_i)}{\sum_k P(\theta_k)P(y_j|\theta_k)}$$

where $P(\theta_j)$ is the prior probability of event θ_i and $P(\theta_j|y_j)$ is the posterior probability of event y_j given θ_i. This is simply a form of Bayes' theorem. Here, $h(\theta)$ is the prior pdf that expresses our belief about the value of θ before the data $(Y = y)$ became available. Then $k(\theta|y)$ is the posterior pdf, given the data $(Y = y)$.

Note that the difference in the shape of the prior pdf $h(\theta)$ compared to the posterior pdf $k(\theta|y)$ due to the information is a result of the product of $g(y|\theta)$ and $h(\theta)$, because $f_l(y)$ is simply a normalization constant for a fixed y. The idea of reliability is to take "prior" data and combine it with current data to gain a better estimate or confidence interval or test than would be possible with either on their own. As more current data is acquired, the prior data is "washed out" [1.1].

1.4 Stochastic Processes

Stochastic processes are used to describe the operation of a system over time. There are two main types of stochastic processes: continuous and discrete. A complex continuous process is a process describing a system transition from state to state. The simplest process that will be discussed here is a Markov process. In this case, the future behavior of the process does not depend on its past or present behavior. In many systems that arise in practice, however, past and present states of the system influence the future states, even if they do not uniquely determine them.

1.4.1 Markov Processes

In this section, we will discuss discrete stochastic processes. As an introduction to the Markov process, let us examine the following example.

Example 1.20: Consider a parallel system consisting of two components. From a reliability point of view, the states of the system can be described by

State 1: Full operation (both components operating);
State 2: One component is operating and one component has failed;
State 3: Both components have failed.

Define

$$P_i(t) = P[X(t) = i] = P[\text{system is in state } i \text{ at time } t]$$

and

$$P_i(t + \mathrm{d}t) = P[X(t + \mathrm{d}t) = i] = P[\text{system is in state } i \text{ at time } t + \mathrm{d}t].$$

Define a random variable $X(t)$ which can assume the values 1, 2, or 3 corresponding to the states mentioned above. Since $X(t)$ is a random variable, one can discuss $P[X(t) = 1]$, $P[X(t) = 2]$ or the conditional probability $P[X(t_1) = 2|X(t_0) = 1]$. Again, $X(t)$ is defined as a function of time t, while the conditional probability $P[X(t_1) = 2|X(t_0) = 1]$ can be interpreted as the probability of being in state 2 at time t_1, given that the system was in state 1 at time t_0. In this example, the "stage space" is discrete, i. e., 1, 2, 3, etc., and the parameter space (time) is continuous. The simple process described above is called a stochastic process: a process that develops over time (or space) in accordance with some probabilistic (stochastic) laws. There are many types of stochastic processes.

Here we emphasize the Markov process, which is a special type of stochastic process. Let the system be observed at discrete moments of time t_n, where $n = 0, 1, 2, \ldots$, and let $X(t_n)$ denote the state of the system at time t_n.

Definition 1.2
Let $t_0 < t_1 < \ldots < t_n$. If

$$P[X(t_n) = x_n|X(t_{n-1}) = x_{n-1}, X(t_{n-2}) = x_{n-2}, \ldots, X(t_0) = x_0] = P[X(t_n) = x_n|X(t_{n-1}) = x_{n-1}]$$

then the process is called a *Markov process*.

From the definition of a Markov process, given the present state of the process, its behavior in the future does not depend on its behavior in the past. Many systems have this property, which is called the *Markov property*, and systems that have this property are called *Markov chains*. The Markov property is precisely defined by the following requirement:

$$P[X(t_n) = x_n|X(t_{n-1}) = x_{n-1}, X(t_{n-2}) = x_{n-2}, \ldots, X(t_0) = x_0] = P[X(t_n) = x_n|X(t_{n-1}) = x_{n-1}].$$

The essential characteristic of a Markov process is that it is a process that has no memory; its future is determined by the present and not the past. If, in addition to having no memory, the process is such that it depends only on the difference $(t + \mathrm{d}t) - t = \mathrm{d}t$ and not the value of t – in other words $P[X(t + \mathrm{d}t) = j|X(t) = i]$ is independent of t – then the process is Markov with stationary transition probabilities or is homogeneous in time. This is the same property noted in exponential event times; in fact, referring back to the graphical representation of $X(t)$, the times between state changes are exponential if the process has stationary transition probabilities.

Thus, a Markov process which is homogeneous in time can describe processes with exponential event occurrence times. The random variable of the process is $X(t)$, the state variable rather than the time to failure used in the exponential failure density. To illustrate the types of processes that can be described, we now review the exponential distribution and its properties. Recall that, if $X_1, X_2, \ldots, X_n$, are independent random variables, each with exponential density and a mean of $1/\lambda_i$, then $\min\{X_1, X_2, \ldots, X_n\}$ has an exponential density with a mean of $\left(\sum \lambda_i\right)^{-1}$.

The significance of this property is as follows:

1. The failure behavior of components operated simultaneously can be characterized by an exponential density with a mean equal to the reciprocal of the sum of the failure rates.
2. The joint failure/repair behavior of a system where components are operating and/or undergoing repair can be characterized by an exponential density with a mean equal to the reciprocal of the sum of the failure and repair rates.
3. The failure/repair behavior of a system similar to that described in (2) above but further complicated by active and dormant operating states and sensing and switching can be characterized by an exponential density.

The above property means that almost all reliability and availability models can be characterized by a time-homogeneous Markov process if the various failure times and repair times are exponential. The notation for the Markov process is $\{X(t), t > 0\}$, where $X(t)$ is discrete (state space) and t is continuous (parameter space). By convention, this type of Markov process is called a continuous-parameter Markov chain.

From a reliability/availability viewpoint, there are two types of Markov processes. These are defined as follows:

1. *Absorbing process:* Contains an "absorbing state", which is a state that, once entered, the system can never leave (e.g. a failure which aborts a flight or a mission).
2. *Ergodic process:* Contains no absorbing states, meaning that $X(t)$ can move around indefinitely (e.g. the operation of a ground power plant where failure only temporarily disrupts the operation).

Pham ([1.1], p. 265) presents a summary of Markov processes broken down into absorbing and ergodic categories. Both the reliability and the availability can be described in terms of the probability of the process or system being in defined "up" states, e.g. states 1 and 2 in the initial example. Likewise, the MTBF can be described as the total time spent in the "up" states before proceeding to the absorbing state or failure state.

Define the incremental transition probability as

$$P_{ij}(\mathrm{d}t) = P[X(t+\mathrm{d}t) = j | X(t) = i].$$

This is the probability that the process [random variable $X(t)$] will move to state j during the increment t to $(t+\mathrm{d}t)$, given that it was in state i at time t. Since we are dealing with time-homogeneous Markov processes (exponential failure and repair times), the incremental transition probabilities can be derived from an analysis of the exponential hazard function. It was shown that the hazard function for an exponential with a mean of $1/\lambda$ was just λ. This means that the limiting (as $\mathrm{d}t \to 0$) conditional probability of an event occurring between t and $t+\mathrm{d}t$, given that an event had not occurred at time t, is simply λ, in other words:

$$h(t) = \lim_{\mathrm{d}t \to 0} \frac{P[t < X < t+\mathrm{d}t | X > t]}{\mathrm{d}t} = \lambda.$$

The equivalent statement for the random variable $X(t)$ is

$$h(t)\,\mathrm{d}t = P[X(t+\mathrm{d}t) = j | X(t) = i] = \lambda\,\mathrm{d}t.$$

Now, $h(t)\,\mathrm{d}t$ is in fact the incremental transition probability, so $P_{ij}(\mathrm{d}t)$ can be stated in terms of the basic failure and/or repair rates.

Returning to Example 1.20, it is easy to construct a state transition showing the incremental transition probabilities between all possible states for the process:

State 1: Both components operating;
State 2: One component up and one component down;
State 3: Both components down (absorbing state).

The loops in *Pham* ([1.1], p. 265) indicate the probability of remaining in the present state during the $\mathrm{d}t$ increment

$$P_{11}(\mathrm{d}t) = 1 - 2\lambda\,\mathrm{d}t \qquad P_{12}(\mathrm{d}t) = 2\lambda\,\mathrm{d}t$$
$$P_{21}(\mathrm{d}t) = 0 \qquad P_{22}(\mathrm{d}t) = 1 - \lambda\,\mathrm{d}t$$
$$P_{31}(\mathrm{d}t) = 0 \qquad P_{32}(\mathrm{d}t) = 0$$

$$P_{13}(\mathrm{d}t) = 0$$
$$P_{23}(\mathrm{d}t) = \lambda\,\mathrm{d}t$$
$$P_{33}(\mathrm{d}t) = 1$$

The zeros for P_{ij}, $i > j$ show that the process cannot go backwards: this is not a repair process. The zero on P_{13} shows that, for a process of this type, the probability of more than one event (e.g. failure, repair, etc.) occurring in the incremental time period $\mathrm{d}t$ approaches zero as $\mathrm{d}t$ approaches zero.

Except for the initial conditions of the process (the state in which the process starts), the process is completely specified by incremental transition probabilities. The reason that this is useful is that assuming exponential event (failure or repair) times allows us to characterize the process at any time t, since the process depends only on what happens between t and $(t+\mathrm{d}t)$. The incremental transition probabilities can be arranged

into a matrix in a way that depicts all possible statewide movements. Thus, for parallel configurations,

$$[p_{ij}(\mathrm{d}t)] = \begin{pmatrix} 1 & 2 & 3 \\ 1-2\lambda\,\mathrm{d}t & 2\lambda\,\mathrm{d}t & 0 \\ 0 & 1-\lambda\,\mathrm{d}t & \lambda\,\mathrm{d}t \\ 0 & 0 & 1 \end{pmatrix}$$

for $i, j = 1, 2$, or 3. The matrix $[P_{ij}(\mathrm{d}t)]$ is called the incremental, one-step transition matrix. It is a stochastic matrix (the rows sum to 1.0). As mentioned earlier, this matrix, along with the initial conditions, completely describes the process.

Now, $[P_{ij}(\mathrm{d}t)]$ gives the probabilities of remaining or moving to all of the various states during the interval t to $t+\mathrm{d}t$; hence,

$$\begin{aligned} P_1(t+\mathrm{d}t) &= (1-2\lambda\,\mathrm{d}t)P_1(t) \\ P_2(t+\mathrm{d}t) &= 2\lambda\,\mathrm{d}tP_1(t)(1-\lambda\,\mathrm{d}t)P_2(t) \\ P_3(t+\mathrm{d}t) &= \lambda\,\mathrm{d}tP_2(t)+P_3(t) \end{aligned}$$

By algebraic manipulation, we have

$$\begin{aligned} \frac{[P_1(t+\mathrm{d}t)-P_1(t)]}{\mathrm{d}t} &= -2\lambda P_1(t), \\ \frac{[P_2(t+\mathrm{d}t)-P_2(t)]}{\mathrm{d}t} &= 2\lambda P_1(t)-\lambda P_2(t), \\ \frac{[P_3(t+\mathrm{d}t)-P_3(t)]}{\mathrm{d}t} &= \lambda P_2(t). \end{aligned}$$

Taking limits of both sides as $\mathrm{d}t \to 0$, we obtain

$$\begin{aligned} P_1'(t) &= -2\lambda P_1(t), \\ P_2'(t) &= 2\lambda P_1(t)-\lambda P_2(t), \\ P_3'(t) &= \lambda P_2(t). \end{aligned}$$

The above system of linear first-order differential equations can be easily solved for $P_1(t)$ and $P_2(t)$, meaning that the reliability of the configuration can be obtained:

$$R(t) = \sum_{i=1}^{2} P_i(t).$$

Actually, there is no need to solve all three equations, only the first two, because $P_3(t)$ does not appear and also $P_3(t) = [1-P_1(t)]-P_2(t)$. The system of linear, first-order differential equations can be solved by various means, including both manual and machine methods. We use manual methods employing the Laplace transform (Appendix B) here.

$$\begin{aligned} L[P_i(t)] &= \int_0^\infty \mathrm{e}^{-st}P_i(t)\,\mathrm{d}t = f_i(s), \\ L[P_i'(t)] &= \int_0^\infty \mathrm{e}^{-st}P_i'(t)\,\mathrm{d}t = sf_i(s)-P_i(0). \end{aligned}$$

Application of the Laplace transform will allow us to transform the system of linear, first-order differential equations into a system of linear algebraic equations that can easily be solved, and solutions of $P_i(t)$ can be determined via the inverse transforms.

Returning to the example, the initial condition of a parallel configuration is assumed to be "fully up", such that

$$P_1(t=0) = 1,\ P_2(t=0) = 0,\ P_3(t=0) = 0.$$

Transforming the equations for $P_1'(t)$ and $P_2'(t)$ gives

$$\begin{aligned} sf_1(s)-P_1(t)|_{t=0} &= -2\lambda f_1(s), \\ sf_2(s)-P_2(t)|_{t=0} &= 2\lambda f_1(s)-\lambda f_2(s). \end{aligned}$$

Evaluating $P_1(t)$ and $P_2(t)$ at $t=0$ gives

$$\begin{aligned} sf_1(s)-1 &= -2\lambda f_1(s), \\ sf_2(s)-0 &= 2\lambda f_1(s)-\lambda f_2(s), \end{aligned}$$

from which we obtain

$$\begin{aligned} (s+2\lambda)f_1(s) &= 1, \\ -2\lambda f_1(s)+(s+\lambda)f_2(s) &= 0. \end{aligned}$$

Solving the above equations for $f_1(s)$ and $f_2(s)$, we have

$$\begin{aligned} f_1(s) &= \frac{1}{(s+2\lambda)}, \\ f_2(s) &= \frac{2\lambda}{[(s+2\lambda)(s+\lambda)]}. \end{aligned}$$

From the inverse Laplace transforms in Appendix B,

$$\begin{aligned} P_1(t) &= \mathrm{e}^{-2\lambda t}, \\ P_2(t) &= 2\mathrm{e}^{-\lambda t}-2\mathrm{e}^{-2\lambda t}, \\ R(t) &= P_1(t)+P_2(t) = 2\mathrm{e}^{-\lambda t}-\mathrm{e}^{-2\lambda t}. \end{aligned}$$

The example given above is that of a simple absorbing process where we are concerned about reliability. If a repair capability were added to the model in the form of a repair rate μ, the methodology would remain the same, with only the final result changing. With

a repair rate μ added to the parallel configuration, the incremental transition matrix would be

$$[P_{ij}(\mathrm{d}t)] = \begin{pmatrix} 1-2\lambda\,\mathrm{d}t & 2\lambda\,\mathrm{d}t & 0 \\ \mu\,\mathrm{d}t & 1-(\lambda+\mu)\,\mathrm{d}t & \lambda\,\mathrm{d}t \\ 0 & 0 & 1 \end{pmatrix}.$$

The differential equations would become

$$P_1'(t) = -2\lambda P_1(t) + \mu P_2(t),$$
$$P_2'(t) = 2\lambda P_1(t) - (\lambda+\mu) P_2(t),$$

and the transformed equations would become

$$(s+2\lambda) f_1(s) - \mu f_2(s) = 1,$$
$$-2\lambda f_1(s) + (s+\lambda+\mu) f_2(s) = 0.$$

Hence, we obtain

$$f_1(s) = \frac{(s+\lambda+\mu)}{(s-s_1)(s-s_2)},$$
$$f_2(s) = \frac{2\lambda}{(s-s_1)(s-s_2)},$$

where

$$s_1 = \frac{-(3\lambda+\mu)+\sqrt{(3\lambda+\mu)2-8\lambda^2}}{2},$$
$$s_2 = \frac{-(3\lambda+\mu)-\sqrt{(3\lambda+\mu)2-8\lambda^2}}{2}. \qquad (1.70)$$

Using the Laplace transform, we obtain

$$P_1(t) = \frac{(s_1+\lambda+\mu)\mathrm{e}^{-s_1 t}}{(s_1-s_2)} + \frac{(s_2+\lambda+\mu)\mathrm{e}^{-s_2 t}}{(s_2-s_1)},$$
$$P_2(t) = \frac{2\lambda\,\mathrm{e}^{-s_1 t}}{(s_1-s_2)} + \frac{2\lambda\,\mathrm{e}^{-s_2 t}}{(s_2-s_1)},$$

where s_1 and s_2 are given in (1.70).

Thus, the reliability of two components in a parallel system is given by

$$R(t) = P_1(t) + P_2(t) = \frac{(s_1+3\lambda+\mu)\mathrm{e}^{-s_1 t} - (s_2+3\lambda+\mu)\mathrm{e}^{-s_2 t}}{(s_1-s_2)} \qquad (1.71)$$

System Mean Time Between Failures

Another parameter of interest for absorbing Markov processes is the MTBF. Recalling the previous example of a parallel configuration with repair, the differential equations $P_1'(t)$ and $P_2'(t)$ describing the process were

$$P_1'(t) = -2\lambda P_1(t) + \mu P_2(t),$$
$$P_2'(t) = 2\lambda P_1(t) - (\lambda+\mu) P_2(t).$$

Integrating both sides of the above equations yields

$$\int_0^\infty P_1'(t)\,\mathrm{d}t = -2\lambda \int_0^\infty P_1(t)\,\mathrm{d}t + \mu \int_0^\infty P_2(t)\,\mathrm{d}t,$$
$$\int_0^\infty P_2'(t)\,\mathrm{d}t = 2\lambda \int_0^\infty P_1(t)\,\mathrm{d}t - (\lambda+\mu) \int_0^\infty P_2(t)\,\mathrm{d}t.$$

For the repairable system and from (1.4),

$$\int_0^\infty R(t)\,\mathrm{d}t = \text{MTBF}.$$

Similarly,

$$\int_0^\infty P_1(t)\,\mathrm{d}t = \text{mean time spent in state 1, and}$$
$$\int_0^\infty P_2(t)\,\mathrm{d}t = \text{mean time spent in state 2.}$$

Designating these mean times as T_1 and T_2, respectively, we have

$$P_1(t)\,\mathrm{d}t|_0^\infty = -2\lambda T_1 + \mu T_2,$$
$$P_2(t)\,\mathrm{d}t|_0^\infty = 2\lambda T_1 - (\lambda+\mu) T_2.$$

But $P_1(t) = 0$ as $t \to \infty$ and $P_1(t) = 1$ for $t = 0$. Likewise, $P_2(t) = 0$ as $t \to \infty$ and $P_2(t) = 0$ for $t = 0$. Thus,

$$-1 = -2\lambda T_1 + \mu T_2,$$
$$0 = 2\lambda T_1 - (\lambda+\mu) T_2,$$

or, equivalently,

$$\begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -2\lambda & \mu \\ 2\lambda & -(\lambda+\mu) \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \end{pmatrix}.$$

Therefore,

$$T_1 = \frac{(\lambda+\mu)}{2\lambda^2}, \quad T_2 = \frac{1}{\lambda},$$
$$\text{MTBF} = T_1 + T_2 = \frac{(\lambda+\mu)}{2\lambda^2} + \frac{1}{\lambda} = \frac{(3\lambda+\mu)}{2\lambda^2}.$$

The MTBF for unmaintained processes is developed in exactly the same way as just shown.

The last case to consider for absorbing processes is that of the availability of a maintained system. The difference between reliability and availability is somewhat subtle for absorbing processes. A good example is that of a communications system where the mission would

continue if such a system failed temporarily, but if it failed permanently the mission would be aborted. Consider a cold-standby system consisting of two units: one main unit and one spare unit [1.1]:

State 1: Main unit operating and the spare is OK;
State 2: Main unit out and restoration underway;
State 3: Spare unit is installed and operating;
State 4: Permanent failure (no spare available).

The incremental transition matrix is given by

$$[P_{ij}(\mathrm{d}t)] = \begin{pmatrix} 1-\lambda\,\mathrm{d}t & \lambda\,\mathrm{d}t & 0 & 0 \\ 0 & 1-\mu\,\mathrm{d}t & \mu\,\mathrm{d}t & 0 \\ 0 & 0 & 1-\lambda\,\mathrm{d}t & \lambda\,\mathrm{d}t \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We obtain

$$P_1'(t) = -\lambda P_1(t),$$
$$P_2'(t) = \lambda P_1(t) - \mu P_2(t),$$
$$P_3'(t) = \mu P_2(t) - \lambda P_3(t).$$

Using the Laplace transform, we obtain the following results.

The probability of full-up performance $P_1(t)$ is given by

$$P_1(t) = \mathrm{e}^{-\lambda t}.$$

The probability of a down system that is under repair $P_2(t)$ is

$$P_2(t) = \left(\frac{\lambda}{(\lambda-\mu)}\right)\left(\mathrm{e}^{-\mu t} - \mathrm{e}^{-\lambda t}\right).$$

Similarly, the probability of a fully up system with no spare available $P_3(t)$ is

$$P_3(t) = \left(\frac{\lambda\mu}{(\lambda-\mu)^2}\right)[\mathrm{e}^{-\mu t} - \mathrm{e}^{-\lambda t} - (\lambda-\mu)t\,\mathrm{e}^{-\lambda t}].$$

Hence, the point availability $A(t)$ is given by

$$A(t) = P_1(t) + P_3(t).$$

If average or interval availability is required, this is achieved by

$$\left(\frac{1}{t}\right)\int_0^T A(t)\,\mathrm{d}t = \left(\frac{1}{t}\right)\int_0^T [P_1(t) + P_3(t)]\,\mathrm{d}t,$$

where T is the interval of concern.

Ergodic processes, as opposed to absorbing processes, do not have any absorbing states, and hence movement between states can go on indefinitely. For the latter reason, availability (point, steady-state, or interval) is the only meaningful measure. As an example of an ergodic process, we will use a ground-based power unit configured in parallel.

The parallel units are identical, each with exponential failure and repair times with means $1/\lambda$ and $1/\mu$, respectively. Assume a two-repairmen capability if required (both units down), then

State 1: Fully up (both units operating);
State 2: One unit down and under repair (other unit up);
State 3: Both units down and under repair.

It should be noted that, as in the case of failure events, two or more repairs cannot be made in the $\mathrm{d}t$ interval.

$$[P_{ij}(\mathrm{d}t)] = \begin{pmatrix} 1-2\lambda\,\mathrm{d}t & 2\lambda\,\mathrm{d}t & 0 \\ \mu\,\mathrm{d}t & 1-(\lambda+\mu)\,\mathrm{d}t & \lambda\,\mathrm{d}t \\ 0 & 2\mu\,\mathrm{d}t & 1-2\mu\,\mathrm{d}t \end{pmatrix}.$$

Case I: *Point Availability – Ergodic Process.* For an ergodic process, as $t \to \infty$ the availability settles down to a constant level. Point availability allows us to study the process before this "settling down", and it reflects the initial conditions in the process. We can obtain a solution for the point availability in a similar way to that for absorbing processes, except that the last row and column of the transition matrix must be retained and entered into the system of equations. For example, the system of differential equations becomes

$$\begin{pmatrix} P_1'(t) \\ P_2'(t) \\ P_3'(t) \end{pmatrix} = \begin{pmatrix} -2\lambda & \mu & 0 \\ 2\lambda & -(\lambda+\mu) & 2\mu \\ 0 & \lambda & -2\mu \end{pmatrix} \begin{pmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{pmatrix}.$$

Similar to the absorbing case, the Laplace transform can be used to solve for $P_1(t)$, $P_2(t)$ and $P_3(t)$; the point availability $A(t)$ is given by

$$A(t) = P_1(t) + P_2(t).$$

Case II: *Interval Availability – Ergodic Process.* This is the same as the absorbing case, with integration over the time period T of interest. The interval availability, $A(T)$, is

$$A(T) = \frac{1}{T}\int_0^T A(t)\,\mathrm{d}t.$$

Case III: *Steady State Availability – Ergodic Process.* Here, the process is examined as $t \to \infty$, with complete "washout" of the initial conditions. By letting $t \to \infty$,

the system of differential equations can be transformed into linear algebraic equations. Thus,

$$\lim_{t\to\infty}\begin{pmatrix}P_1'(t)\\P_2'(t)\\P_3'(t)\end{pmatrix} = \lim_{t\to\infty}\begin{pmatrix}-2\lambda & \mu & 0\\ 2\lambda & -(\lambda+\mu) & 2\mu\\ 0 & \lambda & -2\mu\end{pmatrix}\begin{pmatrix}P_1(t)\\P_2(t)\\P_3(t)\end{pmatrix}.$$

As $t\to\infty$, $P_i(t)\to$ constant and $P_i'(t)\to 0$. This leads to an unsolvable system, namely,

$$\begin{pmatrix}0\\0\\0\end{pmatrix} = \begin{pmatrix}-2\lambda & \mu & 0\\ 2\lambda & -(\lambda+\mu) & 2\mu\\ 0 & \lambda & -2\mu\end{pmatrix}\begin{pmatrix}P_1(t)\\P_2(t)\\P_3(t)\end{pmatrix}.$$

To avoid the above difficulty, an additional equation is introduced:

$$\sum_{i=1}^{3} P_i(t) = 1.$$

With the introduction of the new equation, one of the original equations is deleted and a new system is formed:

$$\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}1 & 1 & 1\\ -2\lambda & \mu & 0\\ 2\lambda & -(\lambda+\mu) & 2\mu\end{pmatrix}\begin{pmatrix}P_1(t)\\P_2(t)\\P_3(t)\end{pmatrix}$$

or, equivalently,

$$\begin{pmatrix}P_1(t)\\P_2(t)\\P_3(t)\end{pmatrix} = \begin{pmatrix}1 & 1 & 1\\ -2\lambda & \mu & 0\\ 2\lambda & -(\lambda+\mu) & 2\mu\end{pmatrix}^{-1}\begin{pmatrix}1\\0\\0\end{pmatrix}.$$

We now obtain the following results:

$$P_1(t) = \frac{\mu^2}{(\mu+\lambda)^2},$$
$$P_2(t) = \frac{2\lambda\mu}{(\mu+\lambda)^2},$$

and

$$P_3(t) = 1 - P_1(t) - P_2(t),$$
$$= \frac{\lambda^2}{(\mu+\lambda)^2}.$$

Therefore, the steady state availability $A(\infty)$ is given by

$$A_3(\infty) = P_1(t) + P_2(t)$$
$$= \frac{\mu(\mu+2\lambda)}{(\mu+\lambda)^2}.$$

Note that Markov methods can also be employed when failure or repair times are not exponential but can be represented as the sum of exponential times with identical means (an Erlang distribution or gamma distribution with integer-valued shape parameters). Basically, the method involves introducing "dummy" states which, although being of no particular interest in themselves, change the hazard function from constant to increasing.

1.4.2 Counting Processes

Among various discrete stochastic processes, counting processes are widely used in engineering statistics to describe the appearance of events in time, such as failures, the number of perfect repairs, etc. The simplest counting process is a Poisson process. The Poisson process plays a special role in many applications related to reliability [1.1]. A classic example of such an application is the decay of uranium. Here, radioactive particles from nuclear material strike a certain target in accordance with a Poisson process of some fixed intensity. One well-known counting process is the so-called renewal process. This process is described as a sequence of events where the intervals between the events are independent and identically distributed random variables. In reliability theory, this type of mathematical model is used to describe the number of occurrences of an event over a time interval. In this section, we also discuss the quasi-renewal process and the nonhomogeneous Poisson process.

A non-negative, integer-valued stochastic process $N(t)$ is called a counting process if $N(t)$ represents the total number of occurrences of an event in the time interval $[0, t]$ and satisfies these two properties:

1. if $t_1 < t_2$, then $N(t_1) \le N(t_2)$,
2. if $t_1 < t_2$, then $N(t_2) - N(t_1)$ is the number of occurrences of the event in the interval $[t_1, t_2]$.

For example, if $N(t)$ equals the number of persons who have entered a restaurant at or prior to time t, then $N(t)$ is a counting process in which an event occurs whenever a person enters the restaurant.

Poisson Processes

One of the most important counting processes is the Poisson process.

Definition 1.3

A counting process $N(t)$ is said to be a Poisson process with intensity λ if

1. the failure process $N(t)$ has stationary independent increments;
2. the number of failures in any time interval of length s has a Poisson distribution with a mean of λs; in other words

$$P\{N(t+s)-N(t)=n\}=\frac{e^{-\lambda s}(\lambda s)^n}{n!} \qquad n=0,1,2,\ldots; \qquad (1.72)$$

3. the initial condition is $N(0)=0$.

This model is also called a homogeneous Poisson process, indicating that the failure rate λ does not depend on time t. In other words, the number of failures that occur during the time interval $(t, t+s]$ does not depend on the current time t, only the length of the time interval s. A counting process is said to possess independent increments if the number of events in disjoint time intervals are independent.

For a stochastic process with independent increments, the autocovariance function is

$$\mathrm{Cov}[X(t_1), X(t_2)] = \begin{cases} \mathrm{Var}[N(t_1+s)-N(t_2)] & \text{for } 0<t_2-t_1<s \\ 0 & \text{otherwise} \end{cases},$$

where

$$X(t)=N(t+s)-N(t).$$

If $X(t)$ is Poisson-distributed, then the variance of the Poisson distribution is

$$\mathrm{Cov}[X(t_1), X(t_2)] = \begin{cases} \lambda[s-(t_2-t_1)] & \text{for } 0<t_2-t_1<s \\ 0 & \text{otherwise} \end{cases}.$$

This result shows that the Poisson increment process is covariance stationary. We now present several properties of the Poisson process.

Property 1.3

The sum of independent Poisson processes $N_1(t)$,$N_2(t)$, $\ldots$, $N_k(t)$ with mean values $\lambda_1 t, \lambda_2 t, \ldots, \lambda_k t$, respectively, is also a Poisson process with mean $\left(\sum_{i=1}^{k}\lambda_i\right)t$. In other words, the sum of the independent Poisson processes is also a Poisson process with a mean that is equal to the sum of the means of the individual Poisson processes.

Property 1.4

The difference between two independent Poisson processes, $N_1(t)$, and $N_2(t)$, with mean $\lambda_1 t$ and $\lambda_2 t$, respectively, is not a Poisson process. Instead, it has a probability mass function of

$$P[N_1(t)-N_2(t)=k] = e^{-(\lambda_1+\lambda_2)t}\left(\frac{\lambda_1}{\lambda_2}\right)^{\frac{k}{2}} I_k\left(2\sqrt{\lambda_1\lambda_2}t\right), \qquad (1.73)$$

where $I_k(.)$ is a modified Bessel function of order k.

Property 1.5

If the Poisson process $N(t)$ with mean λt is filtered such that not every occurrence of the event is counted, then the process has a constant probability p of being counted. The result of this process is a Poisson process with mean $\lambda pt[]$.

Property 1.6

Let $N(t)$ be a Poisson process and Y_n a family of independent and identically distributed random variables which are also independent of $N(t)$. A stochastic process $X(t)$ is said to be a compound Poisson process if it can be represented as

$$X(t)=\sum_{i=1}^{N(t)} Y_i.$$

Renewal Processes

A renewal process is a more general case of the Poisson process in which the inter-arrival times of the process or the times between failures do not necessarily follow the exponential distribution. For convenience, we will call the occurrence of an event a renewal, the inter-arrival time the renewal period, and the waiting time the renewal time.

Definition 1.4

A counting process $N(t)$ that represents the total number of occurrences of an event in the time interval $(0, t]$ is called a renewal process if the times between the failures are independent and identically distributed random variables.

The probability that exactly n failures occur by time t can be written as

$$P[N(t)=n] = P[N(t)\geq n] - P[N(t)>n]. \quad (1.74)$$

Note that the times between the failures are $T_1, T_2, \ldots, T_n$, so the failures occurring at time W_k are

$$W_k = \sum_{i=1}^{k} T_i$$

and

$$T_k = W_k - W_{k-1}.$$

Thus,

$$\begin{aligned} P[N(t)=n] &= P[N(t)\geq n] - P[N(t)>n] \\ &= P[W_n \leq t] - P[W_{n+1} \leq t] \\ &= F_n(t) - F_{n+1}(t), \end{aligned}$$

where $F_n(t)$ is the cumulative distribution function for the time of the nth failure and $n = 0, 1, 2, \ldots$.

Example 1.21: Consider a software testing model for which the time at which an error is found during the testing phase has an exponential distribution with a failure rate of X. It can be shown that the time of the nth failure follows the gamma distribution with parameters k and n. From (1.74), we obtain

$$\begin{aligned} P[N(t)=n] &= P[N(t)\leq n] - P[N(t)\leq n-1] \\ &= \sum_{k=0}^{n} \frac{(\lambda t)^k}{k!} e^{-\lambda t} - \sum_{k=0}^{n-1} \frac{(\lambda t)^k}{k!} e^{-\lambda t} \\ &= \frac{(\lambda t)^n}{n!} e^{-\lambda t} \quad \text{for } n = 0, 1, 2, \ldots. \end{aligned}$$

Several important properties of the renewal function are given below.

Property 1.7

The mean value function of the renewal process, denoted by $m(t)$, is equal to the sum of the distribution functions for all renewal times, that is,

$$\begin{aligned} m(t) &= E[N(t)] \\ &= \sum_{n=1}^{\infty} F_n(t). \end{aligned}$$

Property 1.8

The renewal function $m(t)$ satisfies the following equation:

$$m(t) = F_a(t) + \int_0^t m(t-s)\, dF_a(s), \quad (1.75)$$

where $F_a(t)$ is the distribution function of the interarrival time or the renewal period.

In general, let $y(t)$ be an unknown function to be evaluated and $x(t)$ be any non-negative and integrable function associated with the renewal process. Assume that $F_a(t)$ is the distribution function of the renewal period. We can then obtain the following result.

Property 1.9

Let the renewal equation be

$$y(t) = x(t) + \int_0^t y(t-s)\, dF_a(s). \quad (1.76)$$

Then its solution is given by

$$y(t) = x(t) + \int_0^t x(t-s)\, dm(s),$$

where $m(t)$ is the mean value function of the renewal process.

The proof of the above property can be easily derived using the Laplace transform. Let $x(t) = a$. Thus, in Property 1.9, the solution $y(t)$ is given by

$$\begin{aligned} y(t) &= x(t) + \int_0^t x(t-s)\, dm(s) \\ &= a + \int_0^t a\, dm(s) \\ &= a\{1 + E[N(t)]\}. \end{aligned}$$

Quasi-Renewal Processes

In this section we discuss a general renewal process: the quasi-renewal process. Let $\{N(t), t > 0\}$ be a counting process and let X_n be the time between the $(n-1)$th and the nth event of this process, $n \geq 1$.

Definition 1.5

[1.10]: If the sequence of non-negative random variables $\{X_1, X_2, \dots\}$ is independent and

$$X_i = \alpha X_{i-1} \tag{1.77}$$

for $i \geq 2$ where $\alpha > 0$ is a constant, then the counting process $\{N(t), t \geq 0\}$ is said to be a quasi-renewal process with parameter α and the first inter-arrival time X_1.

When $\alpha = 1$, this process becomes the ordinary renewal process. This quasi-renewal process can be used to model reliability growth processes in software testing phases and hardware burn-in stages for $\alpha > 1$, and in hardware maintenance processes when $\alpha \leq 1$.

Assume that the probability density function, cumulative distribution function, survival function and failure rate of random variable X_1 are $f_1(x)$, $F_1(x)$, $s_1(x)$ and $r_1(x)$, respectively. Then the pfd, cdf, survival function, and failure rate of X_n for $n = 1, 2, 3, \dots$ are, respectively, given below [1.10]:

$$f_n(x) = \frac{1}{\alpha^{n-1}} f_1\left(\frac{1}{\alpha^{n-1}}x\right),$$
$$F_n(x) = F_1\left(\frac{1}{\alpha^{n-1}}x\right),$$
$$s_n(x) = s_1\left(\frac{1}{\alpha^{n-1}}x\right),$$
$$f_n(x) = \frac{1}{\alpha^{n-1}} r_1\left(\frac{1}{\alpha^{n-1}}x\right).$$

Similarly, the mean and variance of X_n is given as

$$E(X_n) = \alpha^{n-1} E(X_1),$$
$$\mathrm{Var}(X_n) = \alpha^{2n-2}\mathrm{Var}(X_1).$$

Because of the non-negativity of X_1, and the fact that X_1 is not identically 0, we obtain

$$E(X_1) = \mu_1 \neq 0.$$

It is worth noting that the shape parameters for X_n are the same for $n = 1, 2, 3, \dots$ for a quasi-renewal process if X_1 follows the gamma, Weibull, or log normal distribution.

This means that the shape parameters of the inter-arrival time will not change after "renewal". In software reliability, the assumption that the software debugging process does not change the error-free distribution seems reasonable. Thus, if a quasi-renewal process model is used, the error-free times that occur during software debugging will have the same shape parameters. In this sense, a quasi-renewal process is suitable for modeling the increase in software reliability. It is worth noting that

$$\lim_{n\to\infty} \frac{E(X_1 + X_2 + \cdots + X_n)}{n} = \lim_{n\to\infty} \frac{\mu_1(1-\alpha^n)}{(1-\alpha)n},$$
$$= 0 \quad \text{if } \alpha < 1,$$
$$= \infty \text{ if } \alpha > 1.$$

Therefore, if the inter-arrival time represents the error-free time of a software system, then the average error-free time approaches infinity when its debugging process has been operating for a long debugging time.

Distribution of N(t). Consider a quasi-renewal process with parameter α and a first inter-arrival time X_1. Clearly, the total number of renewals $N(t)$ that occur up to time t has the following relationship to the arrival time of the nth renewal SS_n:

$$N(t) \geq n \quad \text{if and only if } SS_n \leq t.$$

In other words, $N(t)$ is at least n if and only if the nth renewal occurs prior to time t. It is easily seen that

$$SS_n = \sum_{i=1}^{n} X_i = \sum_{i=1}^{n} \alpha^{i-1} X_1 \quad \text{for } n \geq 1. \tag{1.78}$$

Here, $SS_0 = 0$. Thus, we have

$$P\{N(t) = n\} = P\{N(t) = n\} - P\{N(t) \geq n+1\}$$
$$= P\{SS_n \leq t\} - P\{SS_{n+1} \leq t\}$$
$$= G_n(t) - G_{n+1}(t),$$

where $G_n(t)$ is the convolution of the inter-arrival times $F_1, F_2, F_3, \dots, F_n$. In other words,

$$G_n(t) = P\{F_1 + F_2 + \cdots + F_n \leq t\}.$$

If the mean value of $N(t)$ is defined as the renewal function $m(t)$, then

$$m(t) = E[N(t)]$$
$$= \sum_{n=1}^{\infty} P\{N(t) \geq n\}$$
$$= \sum_{n=1}^{\infty} P\{SS_n \leq t\}$$
$$= \sum_{n=1}^{\infty} G_n(t).$$

The derivative of $m(t)$ is known as the renewal density

$$\lambda(t) = m'(t).$$

In renewal theory, random variables representing inter-arrival distributions assume only non-negative values, and the Laplace transform of its distribution $F_1(t)$ is defined by

$$\mathcal{L}\{F_1(s)\} = \int_0^\infty \mathrm{e}^{-sx}\,\mathrm{d}F_1(x).$$

Therefore,

$$\mathcal{L}F_n(s) = \int_0^\infty \mathrm{e}^{-a^{n-1}st}\,\mathrm{d}F_1(t) = \mathcal{L}F_1(\alpha^{n-1}s)$$

and

$$\begin{aligned}\mathcal{L}m_n(s) &= \sum_{n=1}^{\infty} \mathcal{L}G_n(s)\\ &= \sum_{n=1}^{\infty} \mathcal{L}F_1(s)\mathcal{L}F_1(\alpha s)\cdots\mathcal{L}F_1(\alpha^{n-1}s).\end{aligned}$$

Since there is a one-to-one correspondence between distribution functions and its Laplace transform, it follows that the first inter-arrival distribution of a quasi-renewal process uniquely determines its renewal function.

If the inter-arrival time represents the error-free time (time to first failure), a quasi-renewal process can be used to model reliability growth in both software and hardware.

Suppose that all software faults have the same chance of being detected. If the inter-arrival time of a quasi-renewal process represents the error-free time of a software system, then the expected number of software faults in the time interval $[0, t]$ can be defined by the renewal function, $m(t)$, with parameter $\alpha > 1$. Denoted by $m_\mathrm{r}(t)$, the number of remaining software faults at time t, it follows that

$$m_\mathrm{r}(t) = m(T_\mathrm{c}) - m(t)$$

where $m(T_\mathrm{c})$ is the number of faults that will eventually be detected through a software lifecycle T_c.

Nonhomogeneous Poisson Processes

The nonhomogeneous Poisson process model (NHPP), which represents the number of failures experienced up to time t, is a nonhomogeneous Poisson process $\{N(t)$ with $t \geq 0\}$. The main issue with the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time.

Different assumptions mean that the model will end up with different functional forms of the mean value function. Note that the exponential assumption for the inter-arrival time between failures is relaxed in a renewal process, and the stationary assumption is relaxed in the NHPP.

The NHPP model is based on the following assumptions:

- The failure process has an independent increment; in other words, the number of failures during the time interval $(t, t+s)$ depends on the current time t and the length of the time interval s, and does not depend on the past history of the process.
- The failure rate of the process is given by

$$\begin{aligned}&P\{\text{exactly one failure in } (t, t+\Delta t)\}\\ &\quad = P\{N(t+\Delta t) - N(t) = 1\}\\ &\quad = \lambda(t)\Delta t + o(\Delta t),\end{aligned}$$

  where $\lambda(t)$ is the intensity function.
- During a small interval Δt, the probability of more than one failure is negligible; that is,

$$P\{\text{two or more failures in } (t, t+\Delta t)\} = o(\Delta t),$$

- The initial condition is $N(0) = 0$.

Based on these assumptions, the probability that exactly n failures occur during the time interval $(0, t)$ for the NHPP is given by

$$\Pr\{N(t) = n\} = \frac{[m(t)]^n}{n!}\,\mathrm{e}^{-m(t)} \quad n = 0, 1, 2, \ldots, \tag{1.79}$$

where $m(t) = E[N(t)] = \int_0^t \lambda(s)\mathrm{d}s$ and $\lambda(t)$ is the intensity function. It is easily shown that the mean value function $m(t)$ is nondecreasing.

The reliability $R(t)$, defined as the probability that there are no failures in the time interval $(0, t)$, is given by

$$\begin{aligned}R(t) &= P\{N(t) = 0\}\\ &= \mathrm{e}^{-m(t)}.\end{aligned}$$

In general, the reliability $R(x|t)$ – the probability that there are no failures in the interval $(t, t+x)$ – is given by

$$\begin{aligned}R(x|t) &= P\{N(t+x) - N(t) = 0\}\\ &= \mathrm{e}^{-[m(t+x)-m(t)]}\end{aligned}$$

and its density is given by

$$f(x) = \lambda(t+x)\,\mathrm{e}^{-[m(t+x)-m(t)]},$$

where

$$\lambda(x) = \frac{\partial}{\partial x}[m(x)].$$

The variance of the NHPP can be obtained as follows:

$$\mathrm{Var}[N(t)] = \int_0^t \lambda(s)\,\mathrm{d}s$$

and the autocorrelation function is given by

$$\begin{aligned}\mathrm{Cor}[s] &= E[N(t)]E[N(t+s)-N(t)] + E[N^2(t)] \\ &= \int_0^t \lambda(s)\,\mathrm{d}s \int_0^{t+s} \lambda(s)\,\mathrm{d}s + \int_0^t \lambda(s)\,\mathrm{d}s \\ &= \int_0^t \lambda(s)\,\mathrm{d}s \left[1 + \int_0^{t+s} \lambda(s)\,\mathrm{d}s\right].\end{aligned}$$

Example 1.22: Assume that the intensity λ is a random variable with pdf $f(\lambda)$. Then the probability that exactly n failures occur during the time interval $(0, t)$ is given by

$$P\{N(t) = n\} = \int_0^\infty \mathrm{e}^{-\lambda t}\frac{(\lambda t)^n}{n!} f(\lambda)\,\mathrm{d}\lambda.$$

If the pdf $f(\lambda)$ is given as the following gamma density function with parameters k and m:

$$f(\lambda) = \frac{1}{\Gamma(m)} k^m \lambda^{m-1}\,\mathrm{e}^{-k\lambda} \quad \text{for } \lambda \geq 0$$

then it can be shown that

$$P\{N(t) = n\} = \binom{n+m-1}{n} p^m q^n \quad n = 0, 1, 2, \ldots$$

(this is also called a negative binomial density function), where

$$p = \frac{k}{t+k} \quad \text{and} \quad q = \frac{t}{t+k} = 1 - p.$$

1.5 Further Reading

The reader interested in a deeper understanding of advanced probability theory and stochastic processes should note the following citations, which refer to highly recommended books: [1.9, 11–13]

References

1.1 H. Pham: *Software Reliability* (Springer, Berlin, Heidelberg 2000)

1.2 W. Weibull: A statistical distribution function of wide applicability, J. Appl. Mech. **18**, 293–297 (1951)

1.3 H. Pham: A Vtub-shaped hazard rate function with applications to system safety, Int. J. Reliab. Appl. **3**(1), 1–16 (2002)

1.4 Z. Chen: Exact confidence interval for the shape parameter of a loglogistic distribution, J. Stat. Comput. Sim. **56**, 193–211 (1997)

1.5 W. Nelson: *Applied Life Data Analysis* (Wiley, New York 1982)

1.6 M. Zhao: Statistical reliability change-point estimation models. In: *Handbook of Reliability Engineering*, ed. by H. Pham (Springer, Berlin, Heidelberg 2003) pp. 157–163

1.7 Z. Jelinski, P. B. Moranda: Software reliability research. In: *Statistical Computer Performance Evaluation*, ed. by W. Freiberger (Academic, New York 1972)

1.8 J. D. Musa, A. Lannino, K. Okumoto: *Software Reliability: Measurement, Prediction, and Application* (McGraw-Hill, New York 1987)

1.9 W. Feller: *An Introduction to Probability Theory and Its Applications*, 3rd edn. (Wiley, New York 1994)

1.10 H. Wang, H. Pham: A quasi renewal process and its applications in imperfect maintenance, Int. J. Syst. Sci. **27**(10), 1055–1062 (1996)

1.11 J. L. Devore: *Probability and Statistics for Engineering and the Sciences*, 3rd edn. (Brooks Cole, Pacific Grove 1991)

1.12 B. V. Gnedenko, I. A. Ushakov: *Probabilistic Reliability Engineering* (Wiley, New York 1995)

1.13 J. G. Hahn, W. Q. Meeker: *Statistical Intervals: A Guide for Practitioners* (Wiley, New York 1991)

1.A Appendix: Distribution Tables

Table 1.6 Cumulative areas under the standard normal distribution

Z	0	1	2	3	4	5	6	7	8	9
−3.0	0.0013	0.0010	0.0007	0.0005	0.0003	0.0002	0.0002	0.0001	0.0001	0.0000
−2.9	0.0019	0.0018	0.0017	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
−2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
−2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
−2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
−2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
−2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
−2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
−2.2	0.0139	0.0136	0.0132	0.0129	0.0126	0.0122	0.0119	0.0116	0.0113	0.0110
−2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
−2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
−1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0238	0.0233
−1.8	0.0359	0.0352	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0300	0.0294
−1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
−1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
−1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0570	0.0559
−1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0722	0.0708	0.0694	0.0681
−1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
−1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
−1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
−1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
−0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
−0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
−0.7	0.2420	0.2389	0.2358	0.2327	0.2297	0.2266	0.2236	0.2206	0.2177	0.2148
−0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
−0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
−0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
−0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
−0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
−0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
−0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7703	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015

Table 1.6 (cont.)

Z	0	1	2	3	4	5	6	7	8	9
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9278	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9430	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9648	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9700	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9762	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9874	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9990	0.9993	0.9995	0.9997	0.9998	0.9998	0.9999	0.9999	1.000

Table 1.7 Percentage points for the t-distribution ($t_{\alpha,r}$)

r / α	0.100	0.050	0.025	0.01	0.005	0.0025	0.001
1	3.078	6.314	12.706	31.821	63.657	127.32	318.310
2	1.886	2.920	4.303	6.965	9.925	14.089	23.326
3	1.638	2.353	3.182	4.541	5.841	7.453	10.213
4	1.533	2.132	2.776	3.747	4.604	5.598	7.173
5	1.476	2.015	2.571	3.365	4.032	4.773	5.893
6	1.440	1.943	2.447	3.143	3.707	4.317	5.208
7	1.415	1.895	2.365	2.998	3.499	4.029	4.785
8	1.397	1.860	2.306	2.896	3.355	3.833	4.501
9	1.383	1.833	2.262	2.821	3.250	3.690	4.297
10	1.372	1.812	2.228	2.764	3.169	3.581	4.144
11	1.363	1.796	2.201	2.718	3.106	3.497	4.025
12	1.356	1.782	2.179	2.681	3.055	3.428	3.930
13	1.350	1.771	2.160	2.650	3.012	3.372	3.852
14	1.345	1.761	2.145	2.624	2.977	3.326	3.787
15	1.341	1.753	2.131	2.602	2.947	3.286	3.733
16	1.337	1.746	2.120	2.583	2.921	3.252	3.686
17	1.333	1.740	2.110	2.567	2.898	3.222	3.646
18	1.330	1.734	2.101	2.552	2.878	3.197	3.610
19	1.328	1.729	2.093	2.539	2.861	3.174	3.579
20	1.325	1.725	2.086	2.528	2.845	3.153	3.552
21	1.323	1.721	2.080	2.518	2.831	3.135	3.527
22	1.321	1.717	2.074	2.508	2.819	3.119	3.505
23	1.319	1.714	2.069	2.500	2.807	3.104	3.485
24	1.318	1.711	2.064	2.492	2.797	3.091	3.467

Table 1.7 (cont.)

r / α	0.100	0.050	0.025	0.01	0.005	0.0025	0.001
25	1.316	1.708	2.060	2.485	2.787	3.078	3.450
26	1.315	1.706	2.056	2.479	2.779	3.067	3.435
27	1.314	1.703	2.052	2.473	2.771	3.057	3.421
28	1.313	1.701	2.048	2.467	2.763	3.047	3.408
29	1.311	1.699	2.045	2.462	2.756	3.038	3.396
30	1.310	1.697	2.042	2.457	2.750	3.030	3.385
40	1.303	1.684	2.021	2.423	2.704	2.971	3.307
60	1.296	1.671	2.000	2.390	2.660	2.915	3.232
120	1.289	1.658	1.980	2.358	2.617	2.860	3.160
∞	1.282	1.645	1.960	2.326	2.576	2.807	3.090

Table 1.8 Percentage points for the F-distribution $F_{0.05}$, ν_2/ν_1

ν_2 / ν_1	1	2	3	4	5	6	7	8	9	10
1	161.40	199.50	215.70	224.60	230.20	234.00	236.80	238.90	240.50	241.90
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79
4	7.71	6.94	6.59	6.39	9.26	9.16	6.09	6.04	6.00	5.96
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14
10	4.95	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35
21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32
22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30
23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27
24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24
26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22
27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20
28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19
29	4.18	3.33	2.93	2.70	2.55	2.43	2.35	2.28	2.22	2.18
30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	2.16
40	4.08	3.23	2.84	2.61	2.45	2.34	2.25	2.18	2.12	2.08
60	4.00	3.15	2.76	2.53	2.37	2.25	2.17	2.10	2.04	1.99
120	3.92	3.07	2.68	2.45	2.29	2.17	2.09	2.02	1.96	1.91
∞	3.84	3.00	2.60	2.37	2.21	2.10	2.01	1.94	1.88	1.83

Table 1.9 Percentage points for the χ^2 distribution

ν / χ^2_α	$\chi^2_{0.99}$	$\chi^2_{0.975}$	$\chi^2_{0.95}$	$\chi^2_{0.90}$	$\chi^2_{0.10}$	$\chi^2_{0.05}$	$\chi^2_{0.025}$	$\chi^2_{0.01}$
1	0	0.00	0.00	0.02	2.71	3.84	5.02	6.64
2	0.02	0.05	0.10	0.21	4.61	5.99	7.38	9.21
3	0.12	0.22	0.35	0.58	6.25	7.82	9.35	11.35
4	0.30	0.48	0.71	1.06	7.78	9.49	11.14	13.28
5	0.55	0.83	1.15	1.61	9.24	11.07	12.83	15.09
6	0.87	1.24	1.64	2.20	10.65	12.59	14.45	16.81
7	1.24	1.69	2.17	2.83	12.02	14.07	16.01	18.48
8	1.65	2.18	2.73	3.49	13.36	15.51	17.54	20.09
9	2.09	2.70	3.33	4.17	14.68	16.92	19.02	21.67
10	2.56	3.25	3.94	4.87	15.99	18.31	20.48	23.21
11	3.05	3.82	4.58	5.58	17.28	19.68	21.92	24.73
12	3.57	4.40	5.23	6.30	18.55	21.92	23.34	26.22
13	4.11	5.01	5.89	7.04	19.81	22.36	24.74	27.69
14	4.66	5.63	6.57	7.79	21.06	23.69	26.12	29.14
15	5.23	6.26	7.26	8.57	22.31	25.00	27.49	30.58
16	5.81	6.91	7.96	9.31	23.54	26.30	28.85	32.00
17	6.41	7.56	8.67	10.09	24.77	27.59	30.19	33.41
18	7.02	8.23	9.39	10.87	25.99	28.87	31.53	34.81
19	7.63	8.91	10.12	11.65	27.20	30.14	32.85	36.19
20	8.26	9.59	10.85	12.44	28.41	31.41	34.17	37.57
21	8.90	10.28	11.59	13.24	29.62	32.67	35.48	38.93
22	9.54	10.98	12.34	14.04	30.81	33.92	36.78	40.29
23	10.20	11.69	13.09	14.85	32.01	35.17	38.08	41.64
24	10.86	12.40	13.85	15.66	33.20	36.42	39.36	42.98
25	11.52	13.12	14.61	16.47	34.38	37.65	40.65	44.31
26	12.20	13.84	15.38	17.29	35.56	38.89	41.92	45.64
27	12.88	14.57	16.15	18.11	36.74	40.11	43.19	46.96
28	13.57	15.31	16.93	18.94	37.92	41.34	44.46	48.28
29	14.26	16.05	17.71	19.77	39.09	42.56	45.72	49.59
30	14.95	16.79	18.49	20.60	40.26	43.77	46.98	50.89
35	18.48	20.56	22.46	24.81	46.03	49.80	53.21	57.36
40	22.14	24.42	26.51	29.07	51.78	55.76	59.35	63.71
50	29.69	32.35	34.76	37.71	63.14	67.50	71.42	76.17
60	37.47	40.47	43.19	46.48	74.37	79.08	83.30	88.39
70	45.43	48.75	51.74	55.35	85.50	90.53	95.03	100.44
80	53.53	57.15	60.39	64.30	96.55	101.88	106.63	112.34
90	61.74	65.64	69.12	73.31	107.54	113.15	118.14	124.13
100	70.05	74.22	77.93	82.38	118.47	124.34	129.57	135.81
110	78.45	82.86	86.79	91.50	129.36	135.48	140.92	147.42
120	86.91	91.57	95.70	100.65	140.20	146.57	152.22	158.96

Table 1.10 Critical values $d_{n,\alpha}$ for the Kolmogorov–Smirnov test

n / α	0.2	0.1	0.05	0.02	0.01	n / α	0.2	0.1	0.05	0.02	0.01
1	0.900	0.950	0.975	0.990	0.995	16	0.258	0.295	0.327	0.366	0.392
2	0.684	0.776	0.842	0.900	0.929	17	0.250	0.286	0.318	0.355	0.381
3	0.565	0.636	0.708	0.785	0.829	18	0.244	0.279	0.309	0.346	0.371
4	0.493	0.565	0.624	0.689	0.734	19	0.237	0.271	0.301	0.337	0.361
5	0.447	0.509	0.563	0.627	0.669	20	0.232	0.265	0.294	0.329	0.352
6	0.410	0.468	0.519	0.577	0.617	21	0.226	0.259	0.287	0.321	0.344
7	0.381	0.436	0.483	0.538	0.576	22	0.221	0.253	0.281	0.314	0.337
8	0.358	0.410	0.454	0.507	0.542	23	0.216	0.247	0.275	0.307	0.330
9	0.339	0.387	0.430	0.480	0.513	24	0.212	0.242	0.264	0.301	0.323
10	0.323	0.369	0.409	0.457	0.489	25	0.208	0.238	0.264	0.295	0.317
11	0.308	0.352	0.391	0.437	0.468	26	0.204	0.233	0.259	0.290	0.311
12	0.296	0.338	0.375	0.419	0.449	27	0.200	0.229	0.254	0.284	0.305
13	0.285	0.325	0.361	0.404	0.432	28	0.197	0.225	0.250	0.279	0.300
14	0.275	0.314	0.349	0.390	0.418	29	0.193	0.221	0.246	0.275	0.295
15	0.266	0.304	0.338	0.377	0.404	30	0.190	0.218	0.242	0.270	0.281

1.B Appendix: Laplace Transform

If a function $h(x)$ can be obtained from some prescribed operation on a function $f(x)$, then $h(x)$ is often called a transform of $f(x)$. For example,

$$h(x) = \sqrt{2 + f(x)},$$

$$h(x) = \frac{\partial}{\partial x} f(x).$$

The Laplace transform of $f(t)$ is the function $f^*(s)$, where

$$f^*(s) = \int_0^\infty \mathrm{e}^{-st} f(t)\,\mathrm{d}t.$$

The Laplace transform is often denoted by $f^*(s)$ or $\mathcal{L}(f(t))$ or $\mathcal{L}(f)$. The results of the Laplace transform for a few simple functions are presented below.

Results

1.

$$\mathcal{L}(1) = \int_0^\infty \mathrm{e}^{-st}\,\mathrm{d}t = \frac{1}{s};$$

2.

$$\mathcal{L}(\mathrm{e}^{-at}) = \int_0^\infty \mathrm{e}^{-st}\,\mathrm{e}^{-at}\,\mathrm{d}t = \int_0^\infty \mathrm{e}^{-(s+a)t}\,\mathrm{d}t = \frac{1}{s+a}.$$

3. If $f(t) = \frac{1}{a}\mathrm{e}^{-\frac{t}{a}}$, then

$$\mathcal{L}[f(t)] = \int_0^\infty \mathrm{e}^{-st}\frac{1}{a}\mathrm{e}^{-\frac{t}{a}}\,\mathrm{d}t = \frac{1}{1+sa}.$$

4. If $f(t) = t\,\mathrm{e}^{at}$, then

$$\mathcal{L}[f(t)] = \int_0^\infty \mathrm{e}^{-st}\,t\,\mathrm{e}^{at}\,\mathrm{d}t = \frac{1}{(s-a)^2}.$$

5. If $f(t) = \frac{1}{a}(\mathrm{e}^{at} - 1)$, then

$$\mathcal{L}[f(t)] = \int_0^\infty \mathrm{e}^{-st}\frac{1}{a}(\mathrm{e}^{at} - 1)\,\mathrm{d}t = \frac{1}{s(s-a)}.$$

6. If $f(t) = (1 + at)\mathrm{e}^{at}$, then

$$\mathcal{L}[f(t)] = \int_0^\infty \mathrm{e}^{-st}(1 + at)\mathrm{e}^{at}\,\mathrm{d}t = \frac{s}{(s-a)^2}.$$

Similarly, we can obtain the following results:

7. If $f(t)=\frac{a\mathrm{e}^{at}-b\mathrm{e}^{bt}}{a-b}$, then
$$\mathcal{L}[f(t)]=\frac{s}{(s-a)(s-b)} \quad \text{for } a\neq b.$$
8. If $f(t)=\frac{\alpha^k t^{k-1}\mathrm{e}^{-at}}{\Gamma(k)}$ then
$$\mathcal{L}[f(t)]=\left(\frac{\alpha}{\alpha+s}\right)^k.$$
9. If $f(t)=\frac{\mathrm{e}^{at}-\mathrm{e}^{bt}}{a-b}$, for $a\neq b$, then
$$\mathcal{L}[f(t)]=\frac{1}{(s-a)(s-b)}.$$
10. If $f(t)=\lambda\mathrm{e}^{-\lambda t}$, then
$$\mathcal{L}[f(t)]=\frac{\lambda}{\lambda+s}.$$
11.
$$\begin{aligned}&\mathcal{L}[c_1f_1(t)+c_2f_2(t)]\\&=\int_0^\infty \mathrm{e}^{-st}[c_1f_1(t)+c_2f_2(t)]\,\mathrm{d}t\\&=c_1\mathcal{L}[f_1(t)]+c_2\mathcal{L}[f_2(t)]\,.\end{aligned}$$
12. If $f_i(t)=\lambda_i\mathrm{e}^{-\lambda_i t}$, then
$$\mathcal{L}\left[\sum_{i=1}^n f_i(t)\right]=\sum_{i=1}^n\frac{\lambda_i}{\lambda_i+s}.$$
13.
$$\mathcal{L}\left[\sum_{i=1}^n f_i(t)\right]=\sum_{i=1}^n\mathcal{L}\,[f_i(t)].$$
14.
$$\begin{aligned}\mathcal{L}[f'(t)]&=\int_0^\infty \mathrm{e}^{-st}f'(t)\,\mathrm{d}t\\&=f(t)\mathrm{e}^{-st}\Big|_0^\infty+s\int_0^\infty f(t)\mathrm{e}^{-st}\,\mathrm{d}t\\&=-f(0^+)+sf^*(s)\\&=-f(0^+)+s\mathcal{L}[f(t)].\end{aligned}$$

2. Statistical Reliability with Applications

This chapter reviews fundamental ideas in reliability theory and inference. The first part of the chapter accounts for lifetime distributions that are used in engineering reliability analysis, including general properties of reliability distributions that pertain to lifetime for manufactured products. Certain distributions are formulated on the basis of simple physical properties, and other are more or less empirical. The first part of the chapter ends with a description of graphical and analytical methods to find appropriate lifetime distributions for a set of failure data.

The second part of the chapter describes statistical methods for analyzing reliability data, including maximum likelihood estimation and likelihood ratio testing. Degradation data are more prevalent in experiments in which failure is rare and test time is limited. Special regression techniques for degradation data can be used to draw inference on the underlying lifetime distribution, even if failures are rarely observed.

The last part of the chapter discusses reliability for systems. Along with the components that comprise the system, reliability analysis must take account of the system configuration and (stochastic) component dependencies. System reliability is illustrated with an analysis of logistics systems (e.g., moving goods in a system of product sources and retail outlets). Robust reliability design can be used to construct a supply chain that runs with maximum efficiency or minimum cost.

2.1 Introduction and Literature Review

In every day use, words like reliability and quality have meanings that vary depending on the context. In engineering, *reliability* is defined as the ability of an item to perform its function, usually measured in terms of probability as a function of time. *Quality* denotes how the item conforms to its specifications, so reliability is a measure of the item's quality over time.

Since the time of *Birnbaum* and *Sanders* [2.1], when system reliability emerged as its own discipline, research has centered on the operation of simple systems with identical parts working independently of each other. Today's systems do not fit this mold; system representation must include multifaceted components with several component states that can vacillate between perfect operation and terminal failure. Not only do components interact within systems, but many systems are dynamic in that the system configuration can be expected to change during its operation, perhaps due to component failures or external stresses. Computer software, for example, changes its failure structure during the course of design, testing and implementation.

Statistical methods for reliability analysis grew from this concept of system examination, and system reliability is often gauged through component lifetime testing. This chapter reviews the current framework for statistical reliability and considers some modern needs from experimenters in engineering and the physical sciences.

Statistical analysis of reliability data in engineering applications cannot be summarized comprehensively in a single book chapter such as this. The following books (listed fully in the reference section) serve as an excellent basis for a serious treatment of the subject:

1. Statistical Theory of Reliability and Life Testing by *Barlow* and *Proschan* [2.2]
2. Practical Methods for Reliability Data Analysis by *Ansell* and *Phillips* [2.3]
3. Reliability: Probabilistic Models and Statistical Methods by *Leemis* [2.4]
4. Applied Reliability by *Tobias* and *Trindade* [2.5]
5. Engineering Reliability by *Barlow* [2.6]
6. Reliability for Technology, Engineering and Management by *Kales* [2.7]
7. Statistical Methods for Reliability Data by *Meeker* and *Escobar* [2.8]
8. Reliability Modeling, Prediction, and Optimization by *Blischke* and *Murthy* [2.9]
9. Statistical Methods for the Reliability of Repairable Systems by *Rigdon* and *Basu* [2.10] and
10. Modern Reliability Analysis: A Bayesian Perspective by *Johnson* et al. [2.11]

Some of the books in this list focus on reliability theory, and others focus exclusively on reliability engineering. From the more inclusive books, [2.8] provides a complete, high-level guide to reliability inference tools for an engineer, and most examples have an engineering basis (usually in manufacturing). For reliability problems closely associated with materials testing, *Bogdanoff* and *Kozin* [2.12] connects the physics of degradation to reliability models. *Sobczyk* and *Spencer* [2.13] also relate fatigue to reliability through probability modeling. For reliability prediction in software performance, *Lyu* [2.14] provides a comprehensive guide of engineering procedures for software reliability testing, while a more theoretical alternative by *Singpurwalla* and *Wilson* [2.15] emphasizes probability modeling for software reliability, including hierarchical Bayesian methods. Closely related to reliability modeling in engineering systems, *Bedford* and *Cooke* [2.16] goes over methods of probabilistic risk assessment, which is an integral part of reliability modeling for large and complex systems.

Other texts emphasize reliability assessment in a particular engineering field of interest. For statistical reliability in geotechnical engineering, *Baecher* and *Christian* [2.17] is recommended as it details statistical problems with soil variability, autocorrelation (i. e., Kriging), and load/resistance factors. *Ohring* [2.18] provides a comprehensive guide to reliability assessment for electrical engineering and electronics manufacturing, including reliability pertaining to degradation of contacts (e.g., crack growth in solder), optical-fiber reliability, semiconductor degradation and mass-transport-induced failure. For civil engineering, *Melchers'* [2.19] reliability text has a focus on reliability of structural systems and loads, time-dependent reliability and resistance modeling.

2.2 Lifetime Distributions in Reliability

While engineering research has contributed a great deal of the current methods for reliability life testing, an equally great amount exists in the biological sciences, especially relating to epidemiology and biostatistics. Life testing is a crucial component to both fields, but the bio-related sciences tend to focus on mean lifetimes and numerous risk factors. Engineering methods, on the other hand, are more likely to focus on upper (or lower) percentiles of the lifetime distribution as well as the stochastic dependencies between working components. Another crucial difference between the two research areas is that engineering models are more likely to be based on principles of physics that lead to well-known distributions such as Weibull, log–normal, extreme value and so on.

The failure-time distribution is the most widely used probability tool for modeling product reliability in science and industry. If $f(x)$ represents the probability density function for the product's failure time, then the its reliability is $R(x) = \int_x^\infty f(u)\,\mathrm{d}u$, and $R(t) = 1 - F(t)$ where F is the cumulative distribution function (CDF) corresponding to f. A *quantile* is the CDF's inverse; The p-th quantile of F is the lifetime value t_p such that $F(t_p) = p$. To understand the quality of a manufactured product through these lifetime probability functions, it is often useful to consider the notion of *aging*. For example, the (conditional) reliability of a product that has been working t units of time

is

$$R(x|t) = \frac{R(t+x)}{R(t)}, \text{ if } R(t) > 0\,. \tag{2.1}$$

The rate of change of $R(x|t)$ is an important metric for judging a product's quality, and the conditional *failure rate* function $h(t)$ is defined as

$$h(t) = \lim_{x\to\infty} x^{-1}\frac{R(t)-R(t+x)}{R(t)} = \frac{f(t)}{R(t)}\,. \tag{2.2}$$

The *cumulative failure rate* (sometimes called the *hazard function*) is $H(t) = \int_0^t h(u)\,\mathrm{d}u$, and has many practical uses in reliability theory because of its monotonicity and the fact that $H(t) = -\log R(t)$.

The failure rate clearly communicates how the product ages during different spans of its lifetime. Many manufactured products have an increasing failure rate, but the rate of increase is rarely stable throughout the product's lifetime. If $r(t)$ remains constant, it is easy to show the lifetime distribution is exponential $\left[f(x) = \theta\exp(-\theta x), x > 0\right]$ and the product exhibits no aging characteristics. Many electronic components and other manufactured items have brief initial period when failure rate is relatively high and decrease toward a steady state, where it stays until aging causes the rate to increase. This is called a bath-tub failure rate. The period in which early failures occur (called *infant mortality*) is called the *burn-in* period, and is often used by manufacturers to age products and filter out defectives (early failures) before being making it available to the consumer.

2.2.1 Alternative Properties to Describe Reliability

The failure rate function, reliability function, cumulative hazard function, and probability density describe different aspects of a lifetime distribution. The expected lifetime, or *mean time to failure* (MTTF) is an important measure for repairable systems. Several alternatives for characterizing properties of the lifetime distribution include:

- *Mean residual life* $= L(t) = E_X(X - t|X \geq t)$ is the expected residual life of a component that has already lasted t units of time. If $L(t)$ is less than the expected lifetime μ, the product is exhibiting aging by the time t.
- *Reversed hazard rate* $= \nu(t) = f(x)/F(x)$ provides a different aspect of reliability: the conditional failure frequency at the time just before t given that the product failed in $(0, t]$ (see Chapt. 1 of [2.20], for example).
- *Percentile residual life* $= Q_\alpha = F^{-1}\left[1-(1-\alpha)\times R(t)\right] - t$ is the α quantile of the residual life (the conditional lifetime distribution given that the product has lasted t units of time). The median residual life, where $\alpha = 1/2$ compares closely to $L(t)$.
- *Mill's ratio* $= R(x)/f(x) = 1/h(x)$, used in economics, is not an ordinary way to characterize reliability, but it is worth noting because of its close connection to failure rate.

2.2.2 Conventional Reliability Lifetime Distributions

So far, only one distribution (exponential) has been mentioned. Rather than presenting a formal review of commonly used reliability distributions, a summary of commonly applied lifetime distributions is presented in Table 2.1, including the exponential, gamma, Weibull, log–normal, logistic, Pareto and extreme value. In the table, $\Gamma(t) = \int_0^\infty x^{t-1}\,\mathrm{e}^{-x}\,\mathrm{d}x$ is the ordinary gamma function, and $IG(t, x)$ represents the corresponding incomplete Gamma function.

For manufacturing centers and research laboratories that conduct lifetime tests on products, lifetime data is an essential element of reliability analysis. However, a great deal of reliability analysis is based on field data, or reliability information sampled from day-to-day usage of the product. In many of these instances, lifetime data is a luxury not afforded to the reliability inference. Instead, historical event data and inspection counts are logged for the data analysis. Consequently, several discrete distributions (e.g., Poisson, binomial, geometric) are important in reliability applications. Chapter 4 has a more detailed discussion of these and other statistical distributions applied in engineering problems.

2.2.3 From Physics to Failure Distributions

Many of the distributions in Table 2.1 are derived based on physical principles. For example, *Weibull* [2.21] derived the distribution that takes his name to represent the breaking strength of materials based on the idea that some components are comparable to a chain that is no stronger than its weakest link. From this premise, the distribution can be derived from properties of minimums, in contrast to the extreme value distribution, which can be derived through the properties of maximums (see [2.22], for example). In a short time after its introduction, the Weibull distribution was successfully applied to numerous modeling problems in engineering and has become the hallmark distribution in applied re-

Table 2.1 Common lifetime distributions used in reliability data analysis

Distribution	$f(t),\quad t>0$	$h(t)$	μ	σ^2	Parameter space
Exponential	$\theta\,\mathrm{e}^{-\theta t}$	θ	$1/\theta$	$1/\theta^2$	$\theta>0$
Weibull	$\lambda\kappa t^{\kappa-1}\,\mathrm{e}^{-\lambda t^{\kappa}}$	$\lambda\kappa t^{\kappa-1}$	$\lambda^{-1/\kappa}\Gamma\left(1+\frac{1}{\kappa}\right)$	$\lambda^{-2/\kappa}\left[\Gamma\left(1+\frac{2}{\kappa}\right)-\Gamma^2\left(1+\frac{1}{\kappa}\right)\right]$	$\kappa>0,\lambda>0$
Gamma	$\lambda^r\Gamma^{-1}(r)t^{r-1}\,\mathrm{e}^{-\lambda t}$	$\frac{\lambda^r t^{r-1}\,\mathrm{e}^{-\lambda t}}{\Gamma(r)\,[1-IG(r,\lambda t)]}$	r/λ	r/λ^2	$r>0,\lambda>0$
Log–normal	$\frac{1}{\sigma\sqrt{2\pi}}\,\mathrm{e}^{\frac{-(\log t-\mu)^2}{2\sigma^2}}$	$f(t)/R(t)$	$\mathrm{e}^{\mu+\sigma^2/2}$	$\mathrm{e}^{2\mu+2\sigma^2}-\mathrm{e}^{2\mu+\sigma^2}$	$-\infty<\mu<\infty,\sigma>0$
Logistic	$\frac{\mathrm{e}^{-(t-\lambda)/\beta}}{\beta\left(1+\mathrm{e}^{-(t-\lambda)/\beta}\right)^2}$	$\left[\beta\left(1+\mathrm{e}^{-(t-\lambda)/\beta}\right)\right]^{-1}$	λ	$(\beta\pi)^2/3$	$-\infty<\lambda<\infty,\beta>0$
Pareto	$\frac{m\theta^m}{t^{m+1}}$	$\frac{m}{t}$	$\frac{m\theta}{m-1}$	$\frac{m\theta^2}{(m-1)^2(m-2)}$	$t>\theta,m>0$
Extreme value	$\frac{\exp[-(t-a)/b]}{b\exp[-\exp(-(t-a)/b)]}$	$\frac{\exp[-(t-a)/b]}{b\exp[-\exp(-(t-a)/b)]-1}$	$a-b\Gamma'(1)$	$(b\pi)^2/6$	$-\infty<a<\infty,b>0$

liability. A primary reason for its suitability to lifetime analysis is its flexible failure rate; unlike other distributions listed in Table 2.1, the Weibull failure rate is simple to model, easy to demonstrate and it can be either increasing or decreasing. A mixture of two Weibull distributions can be used to portray a bath-tub failure rate (as long as only one of the shape parameters is less than one). *Mudholkar* et al. [2.23] introduce a new shape parameter to a generalized Weibull distribution that allows bath-tub-shaped failure rates as well as a broader class of monotone failure rates.

For materials exposed to constant stress cycles with a given stress range, lifetime is measured in number of cycles until failure (N). The Whöler curve (or S–N curve) relates stress level (S) to N as $NS^b=k$, where b and k are material parameters (see [2.13] for examples). By taking logarithms of the S–N equation, we can express cycles to failure as a linear function: $Y=\log N=\log k-b\log S$. If N is log–normally distributed, then Y is normally distributed and regular regression models can be applied for predicting cycles to failure (at a given stress level). In many settings, the log–normal distribution is applied as the failure time distribution when the corresponding degradation process based on rates that combine multiplicatively. Despite having a concave-shaped (or upside-down bath-tub shape) failure rate, the log–normal is especially useful in modeling fatigue crack growth in metals and composites.

Birnbaum and *Saunders* [2.1] modeled the damage to a test item after n cycles as $B_n=\zeta_1+\cdots+\zeta_n$, where ζ_i represents the damage amassed in the i-th cycle. If failure is determined by B_n exceeding a fixed damage threshold value B^*, and if the ζ_i are identically and independently distributed,

$$P(N\le n)=P(B_n>B^*)\approx\Phi\left(\frac{B^*-n\mu}{\sigma\sqrt{n}}\right)\,, \tag{2.3}$$

where Φ is the standard normal CDF. This results because B_n will be approximately normal if n is large enough. The reliability function for the test unit is

$$R(t)\approx\Phi\left(\frac{B^*-n\mu}{\sigma\sqrt{n}}\right) \tag{2.4}$$

which is called the *Birnbaum–Saunders* distribution. It follows that

$$W=\frac{\mu\sqrt{N}}{\sigma}-\frac{B^*}{\sigma\sqrt{N}} \tag{2.5}$$

has a normal distribution, which leads to accessible implementation in lifetime modeling (see [2.24] or [2.12] for more properties).

2.2.4 Lifetime Distributions from Degradation Modeling

These examples show how the product's lifetime distribution can be implied by knowledge of how it degrades in time. In general, degradation measurements have great potential to improve lifetime data analysis, but they also introduce new problems to the statistical inference. Lifetime models have been researched and refined for

many manufactured products that are put on test. On the other hand, degradation models tend to be empirical (e.g., nonparametric) or based on simple physical properties of the test item and its environment (e.g., the Paris crack law, Arrhenius rule, power law) which often lead to obscure lifetime models. *Meeker* and *Escobar* [2.8] provide a comprehensive guide to degradation modeling, and show that many valid degradation models will not yield lifetime distributions with closed-form solutions. Given the improving computational tools available to researchers, this is no deterrent to using degradation analysis; users of the S-plus programming software can access advanced tools for degradation analysis from SPLIDA (S-plus functions for life data analysis) developed by *Meeker* [2.25].

In a setting where the lifetime distribution is known, but the degradation distribution is unknown, degradation information does not necessarily complement the available lifetime data. For example, the lifetime data may be distributed as Weibull, but conventional degradation models will contradict the Weibull assumption (actually, the rarely used *reciprocal Weibull* distribution for degradation with a fixed failure threshold leads to Weibull lifetimes).

In selecting a degradation model based on longitudinal measurements of degradation, monotonic models are typically chosen under the assumption that degradation is a one-way process. In some cases, such as the measured luminosity of light displays (vacuum fluorescent displays, plasma display devices), the degradation is not necessarily monotonic because, during the first phase of product life, impurities inside the light display's vacuum are slowly burned off and luminosity increases. After achieving a peak level, usually before 100 hours of use, the light slowly degrades in a generally monotonic fashion. See *Bae* and *Kvam* [2.26, 27] for details on the modeling of non-monotonic degradation data. Degradation data analysis is summarized in Sect. 2.3.3.

2.2.5 Censoring

For most products tested in regular-use conditions (as opposed to especially harsh conditions), the allotted test time is usually too short to allow the experimenter to witness failure times for the entire set that is on test. When the item is necessarily taken off test after a certain amount of test time, its lifetime is *right censored*. This is also called type I censoring. Type II censoring corresponds to tests that are stopped after a certain number of failures (say k out of n, $1 \le k \le n$) occur.

Inspection data are lifetimes only observed at fixed times of inspection. If the inspection reveals a failed test item, it must be *left censored* at that fixed time. Items that are still working at the time of the last inspection are necessarily right censored. This is sometimes called *interval censoring*.

Censoring is a common hindrance in engineering applications. Lifetime data that are eclipsed by censoring cause serious problems in the data analysis, but it must be kept in mind that each observation, censored or not, contributes information and increases precision in the statistical inference, overall.

2.2.6 Probability Plotting

Probability plotting is a practical tool for checking the adequacy of a fitted lifetime distribution to a given set of data. The rationale is to transform the observed data according to a given distribution so a linear relationship exists if the distribution was specified correctly. In the past, probability plotting paper was employed to construct the transformation, but researchers can find plotting options on many computer packages that feature data analysis (e.g., SAS, S-Plus, Matlab, Minitab, SPSS) making the special plotting paper nearly obsolete. Despite the applicability of this technique, few engineering texts feature in-depth discussion on probability plotting and statistics texts tend to focus on theory more than implementation. *Rigdon* and *Basu* [2.10] provide a thorough discussion of basic probability plotting, and *Atkinson* [2.28] provides a substantial discussion of the subject in the context of regression diagnostics. Advanced plotting techniques even allow for censored observations (see *Waller* and *Turnbull* [2.29], for example).

To illustrate how the plot works, we first linearize the CDF of the distribution in question. For example, if we consider the two-parameter Weibull distribution, the *quantile function* is

$$t_p = \left(\frac{-\log p}{\lambda}\right)^{1/\kappa}, \tag{2.6}$$

which implies that the plot of $\log t$ has a linear relationship with the log–log function of $p = F(t)$. Hence, Weibull probability plots are graphed on log–log probability paper. Figure 2.1 shows a Weibull plot (using Minitab) for the fatigue life of 67 alloy specimens that failed before $n = 300\,000$ cycles. This data set is from *Meeker* and *Escobar* [2.8] and the plot also includes 95% confidence bands that identify the uncertainty associated with the plot. In this case the curvature (especially

2.3 Analysis of Reliability Data

Once the lifetime distribution of a test item is determined, the data can be used to estimate important properties of the distribution, including mean, standard deviation, failure rate, reliability (at a fixed time t) and upper or lower quantiles that pertain to early or late failure times.

There are two fundamental methods for approaching the analysis of lifetime data: Bayesian methods and, for the lack of an optimal term, non-Bayesian methods. Although Bayesian methods are accepted widely across many fields of engineering and physical science, non-Bayesian statistics, mostly frequentist and likelihood methods, are still an industry standard. This chapter will not detail how methods of statistical inference are derived in various frameworks of statistical ideology. Accelerated life testing, an important tool for designing reliability experiments, is discussed in detail in Chapt. 22 and is only mentioned in this chapter. Instead, a summary of important procedures is outlined for statistical estimation, confidence intervals and hypothesis tests.

2.3.1 Maximum Likelihood

Parametric likelihood methods examine a family of probability distributions and choose the parameter combination that best fits the data. A likelihood function is generally defined by the observed probability model; if the lifetime data $X_1, \ldots, X_n$ are independently and identically (i.i.d.) distributed with density function $f_X(x;\theta)$, the likelihood function is

$$L(\theta) = \prod_{i=1}^{n} f_X(x_i;\theta) \tag{2.7}$$

and the *maximum likelihood estimator* (MLE) is the value of θ that maximizes $L(\theta)$. Single-parameter distributions such as the exponential generate easily solved MLEs, but distributions with two or more parameters are not often straightforward. Samples that are not IID lead to complicated likelihood functions and numerical methods are usually employed to solve for MLEs. If an observation x represents a right-censoring time, for example, then $P(\text{censor}) = R(x)$ and this information contributes the term $R(x)$ to the likelihood instead of $f(x)$. *Leemis* [2.4] provides a thorough introduction to likelihood theory for reliability inference.

For most parametric distributions of interest, the MLE $(\hat{\theta})$ has helpful limit properties. As the sample size $n \to \infty$, $\sqrt{n}(\hat{\theta} - \theta) \to N\left[0, i(\theta)^{-1}\right]$, where

$$i(\theta) = \mathrm{E}\left[\left(\frac{\partial}{\partial\theta}\log f\right)^2\right] = -\mathrm{E}\left(\frac{\partial^2}{\partial\theta^2}\log f\right) \tag{2.8}$$

is the estimator's *Fisher information*. For other parameters of interest, say $\psi(\theta)$, we can construct approximate confidence intervals based on an estimated variance using the Fisher information:

$$\hat{\sigma}^2\left[\psi(\hat{\theta})\right] \approx \frac{1}{\psi(\hat{\theta})^2 i(\hat{\theta})} \,. \tag{2.9}$$

This allows the analyst to make direct inference for the component reliability $\left[\psi(\theta;t) = R_\theta(t)$, for example$\right]$.

Example. MLE for failure rate with exponential data $(X_1, \ldots, X_n)$: the likelihood is based on $f(x) = \theta\exp(-\theta x)$ and is easier to maximize in its natural-log form

$$\log L(\theta) = \log\left(\prod_{i=1}^{n}\theta\,\mathrm{e}^{-\theta x_i}\right) = n\log\theta - \theta\sum_{i=1}^{n} x_i \,.$$

The maximum occurs at $\hat{\theta} = 1/\bar{x}$, and the Fisher information $i(\theta) = n/\theta^2$, so an approximate $(1-\alpha)$ confidence interval is

$$\frac{1}{\bar{x}} \pm z_{\frac{\alpha}{2}}\, i(\hat{\theta})^{-1/2} = \frac{1}{\bar{x}} \pm z_{\frac{\alpha}{2}}\frac{\hat{\theta}}{\sqrt{n}} = \frac{1}{\bar{x}} \pm z_{\frac{\alpha}{2}}\left(\bar{x}\sqrt{n}\right)^{-1} . \tag{2.10}$$

In this case, the approximation above is surpassed by an exact interval that can be constructed from the statistic $2\theta(X_1 + \cdots + X_n)$ which has a chi-squared distribution with $2n$ degrees of freedom. The confidence statement $P\left[\chi^2_{2n}(1-\alpha/2) \le (X_1 + \cdots + X_n) \le \chi^2_{2n}(\alpha/2)\right] = 1-\alpha$, where $\chi^2_{2n}(\alpha)$ represents the α quantile of the chi-squared distribution with $2n$ degrees of freedom, leads to a $1-\alpha$ confidence interval for θ of

$$\left(\frac{\chi^2_{2n}(1-\alpha/2)}{2n\bar{x}}, \frac{\chi^2_{2n}(\alpha/2)}{2n\bar{x}}\right) . \tag{2.11}$$

2.3.2 Likelihood Ratio

Uncertainty bounds, especially for multidimensional parameters, are more directly computed using the likelihood ratio (LR) method. Here we consider θ to have p components. Confidence regions are constructed by

actual contours (in p-dimensions) of the likelihood function. Define the LR as

$$\Lambda(\theta, \hat{\theta}) = \frac{L(\theta)}{L(\hat{\theta})} , \tag{2.12}$$

where $\hat{\theta}$ is the MLE of L. If θ is the true value of the parameter, then

$$-2 \log \Lambda \sim \chi_p^2 ,$$

where χ_p^2 is the chi-squared distribution with p degrees of freedom. A $(1-\alpha)$ confidence region for θ is

$$\left\{ \theta : -2 \log \Lambda(\theta, \hat{\theta}) \le \chi_p^2(\alpha) \right\} , \tag{2.13}$$

where $\chi_p^2(\alpha)$ represents the $1-\alpha$ quantile of the χ_p^2 distribution.

Example. Confidence region for Weibull parameters: In this case, the MLEs for $\theta = (\lambda, r)$ must be computed using numerical methods. Many statistical software packages compute such estimators along with confidence bounds. With $(\hat{\lambda}, \hat{r})$, $L(\hat{\lambda}, \hat{r})$ standardizes the likelihood ratio so that $0 \le \Lambda(\theta, \hat{\theta}) \le 1$ and Λ peaks at $(\lambda, r) = (\hat{\lambda}, \hat{r})$. Figure 2.2 shows 50%, 90% and 95% confidence regions for the Weibull parameters based on a simulated sample of $n = 100$.

Empirical likelihood provides a powerful method for providing confidence bounds on parameters of inference without necessarily making strong assumptions about the lifetime distribution of the product (i. e., it is *nonparametric*). This chapter cannot afford the space needed to provide the reader with an adequate description of its method and theory; *Owen* [2.30] provides a comprehensive study of empirical likelihood including its application to lifetime data.

2.3.3 Degradation Data

As an alternative to traditional life testing, degradation tests can be effective in assessing product reliability when measurements of degradation leading to failure are observable and quantifiable. *Meeker* and *Escobar* [2.8] provide the most comprehensive discussion on modeling and analyzing degradation data for manufactured items that have either a soft failure threshold (i. e., an arbitrary fixed point at which the device is considered to have failed) or items that degrade before reaching a failed state. In the electronics industry, product lifetimes are far too long to test in a laboratory; some products in the lab will tend to become obsolete long before they actually fail. In such cases, accelerated degradation testing (ADT) is used to hasten product failure. In the manufacture of electronic components, this is often accomplished by increasing voltage or temperature. See Chapt. 22 for a review of recent results in ALT.

If the degradation path is modeled as

$$y_i(t) = \eta_i(t) + \epsilon_i(t) , \tag{2.14}$$

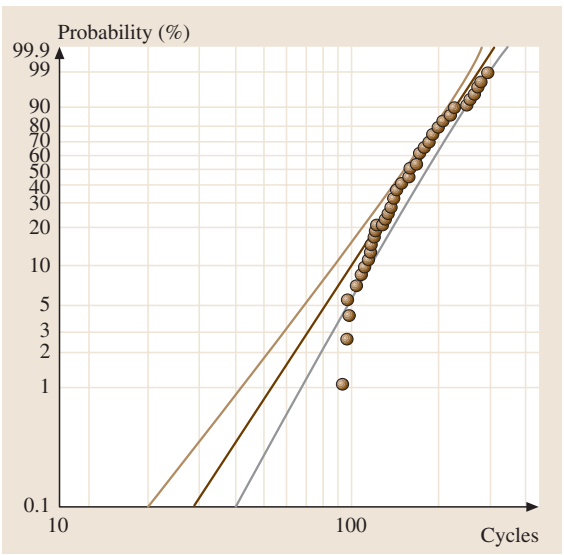


Fig. 2.1 Weibull probability plot for alloy T7987 fatigue life [2.8]

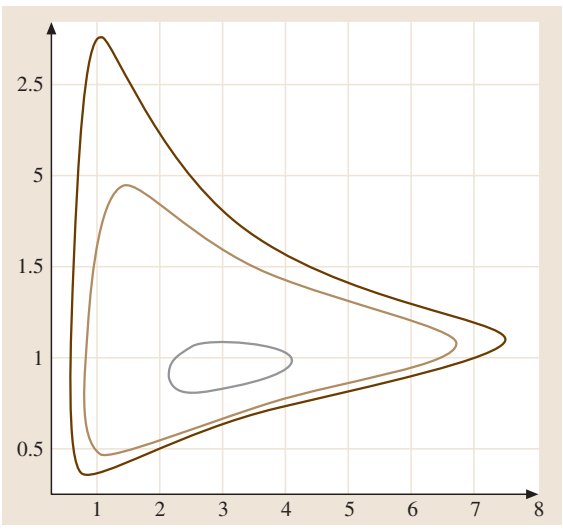


Fig. 2.2 $1-\alpha = 0.50, 0.90, 0.95$ confidence regions for Weibull parameters (λ, r) based on simulated data of size $n = 100$

where η_i is the path of the i-th tested unit ($i = 1, \ldots, n$) and ϵ_i represents an error term that has a distribution $H(\epsilon; \Sigma)$ with parameter Σ unknown. Failure would be declared once $y_i(t)$ passes a certain degradation threshold, say y^*. The lifetime distribution can be computed as (assuming degradation is an increasing function)

$$F(t) = P\left[y(t) > y^*\right] = P\left[\epsilon_i(t) > y^* - \eta_i(t)\right] . \tag{2.15}$$

If η is a deterministic function, the lifetime distribution is driven completely by the error term. This is not altogether realistic. In most cases, item-to-item variability exists and the function η contains *random coefficients*; that is, $\eta(t) = \eta(t, \lambda, \theta)$, where λ is a vector of unknown parameters (common to all units) and θ is a vector of random coefficients which have a distribution G (with further unknown parameters β) so that realizations of θ change from unit to unit. With an accumulated set of unknown parameters (λ, β, Σ), this makes for a difficult computation of the lifetime distribution. Numerical methods and simulations are typically employed to generate point estimates and confidence statements.

Least squares or maximum likelihood can be used to estimate the unknown parameters in the degradation model. To estimate $F(t_0)$, one can simulate M degradation curves (choosing M to be large) from the estimated regression by generating M random coefficients $\theta_1, \ldots, \theta_M$ from the estimated distribution $G(\theta; \hat{\beta})$. Next compute the estimated degradation curve for y_i based on the model with θ_i and $\hat{\lambda}$: $y_i(t) = \eta_i(t; \hat{\lambda}, \theta_i)$. Then $\hat{F}(t_0)$ is the proportion of the M generated curves that have reached the failure threshold y^* by time t_0.

Meeker and Escobar use *bootstrap confidence intervals* for measuring the uncertainty in the lifetime distribution estimate. Their method follows the general algorithm for nonparametric bootstrap confidence intervals described in *Efron* and *Tibshirani* [2.31]. There are numerous bootstrap sampling methods for various uncertainty problems posed by complex models. This algorithm uses a nonparametric bootstrap sampling procedure which *resamples* n of the sample degradation curves *with replacement* (i. e., so some curves may not be represented in the sample while others may be represented multiple times). This resampled set will be termed the bootstrap sample in the following procedure for constructing confidence intervals.

1. Compute estimates of the parameters β, λ, Σ.
2. Use simulation (as above) to construct $\hat{F}(t_0)$.
3. Generate $N \geq 1000$ bootstrap samples, and for each one, compute estimates $\hat{F}^{(1)}(t_0), \ldots, \hat{F}^{(N)}(t_0)$. This is done as before except now the M simulated degradation paths are constructed with an error term generated from $H(\eta; \hat{\Sigma})$ to reflect variability in any single degradation path.
4. With the collection of bootstrap estimates from step 3, compute a $1 - \alpha$ confidence interval for $F(t_0)$ as $\left[\hat{F}^l(t_0), \hat{F}^u(t_0)\right]$, where the indexes $1 \leq l \leq u \leq N$ are calculated as $l/N = \Phi[2\Phi^{-1/2}(p_0) + \Phi^{-1/2} \times (\alpha/2)]$ and $u/N = \Phi[2\Phi^{-1/2}(p_0) + \Phi^{-1/2} \times (1 - \alpha/2)]$, and p_0 is the proportion of bootstrap estimates of $F(t_0)$ less than $\hat{F}(t_0)$.

Procedures based on realistic degradation models can obviously grow to be computationally cumbersome, but for important applications the increase in statistical efficiency can be dramatic. In the past, these computations have impeded degradation analysis from being a feature of reliability problem solving. Such analyses are easier to implement now, and the reliability analyst need not be coerced into using an overly simplistic model—for instance, a linear model that does not allow for random coefficients.

2.4 System Reliability

A *system* is an arrangement of components that work together for a common goal. So far, the discussion has fixated on the lifetime analysis of a single component, so this represents an extension of single-component reliability study. At the simplest level, a system contains n components of an identical type that are assumed to function independently. The mapping of component outcomes to system outcomes is through the system's structure function. The *reliability function* describes the system reliability as a function of component reliability.

A *series* system is such that the failure of any of the n components in the working group causes the system to fail. If the probability that a single component fails in its mission is p, the probability the system fails is $1 - P(\text{system succeeds}) = 1 - P(\text{all } n \text{ components succeed}) = 1 - (1 - p)^n$. More generally, in terms of component reliabili-

ties $(p_1, \ldots, p_n)$, the system reliability function Ψ is

$$\Psi(p_1, \ldots, p_n) = \prod_{i=1}^{n} (1 - p_i) \,. \tag{2.16}$$

A *parallel* system is just the opposite; it fails only after every one of its n working components fail. The system failure probability is then

$$\Psi(p_1, \ldots, p_n) = 1 - \prod_{i=1}^{n} p_i \,. \tag{2.17}$$

The parallel system and series system are special cases of a k-out-of-n system, which is a system that works as long as at least k out of its n components work. Assuming $p_i = p, i = 1, \ldots, n$, the reliability of a k-out-of-n systems is

$$\Psi(p) = \sum_{i=k}^{n} \binom{n}{i} (1-p)^i p^{n-1} \,. \tag{2.18}$$

Of course, most component arrangements are much more complex that a series or parallel system. With just three components, there are five unique ways of arranging the components in a coherent way (that is, so that each component success contributes positively to the system reliability). Figure 2.3 shows the *system structure* of those five arrangements in terms of a *logic diagram* including a series system (1), a 2-out-of-3 system (3), and a parallel system (5). Note that the 2-out-of-3 system cannot be diagrammed with only three components, so each component is represented twice in the logic diagram. Figure 2.4 displays the corresponding reliabilities, as a function of the component reliability $0 \le p \le 1$ of those five systems. Fundamental properties of coherent systems are discussed in [2.2] and [2.4].

2.4.1 Estimating System and Component Reliability

In many complex systems, the reliability of the system can be computed through the reliability of the components along with the system's structure function. If the exact reliability is too difficult to compute explicitly, reliability bounds might be achievable based on *minimum cut sets* (MCS) and *minimum path sets* (MPS). An MPS is the collection of the smallest component sets that are required to work in order to keep the system working. An MCS is the collection of the smallest component sets that are required to fail in order for the system to fail. Table 2.2 shows the minimum cuts sets and path sets for the three-component systems from Fig. 2.3.

In most industrial systems, components have different roles and varying reliabilities, and often the component reliability depends on the working status of other components. System reliability can be simplified through fault-tree analyses (see Chapt. 7 of [2.16], for example), but uncertainty bounds for system reliability are typically determined through simulation.

In laboratory tests, component reliabilities are determined and the system reliability is computed as a function of the statistical inference of component lifetimes. In field studies, the tables are turned. Component manufacturers seeking reliability data outside laboratory

Table 2.2 Minimum cut sets and path sets for the systems in Fig. 2.3

System	Minimum path sets	Minimum cut sets
1	{A,B,C}	{A}, {B}, {C}
2	{A,B}, {C}	{A,C}, {B,C}
3	{A,B}, {A,C}, {B,C}	{A,B}, {A,C}, {B,C}
4	{A,B}, {A,C}	{A}, {B,C}
5	{A}, {B}, {C}	{A,B,C}

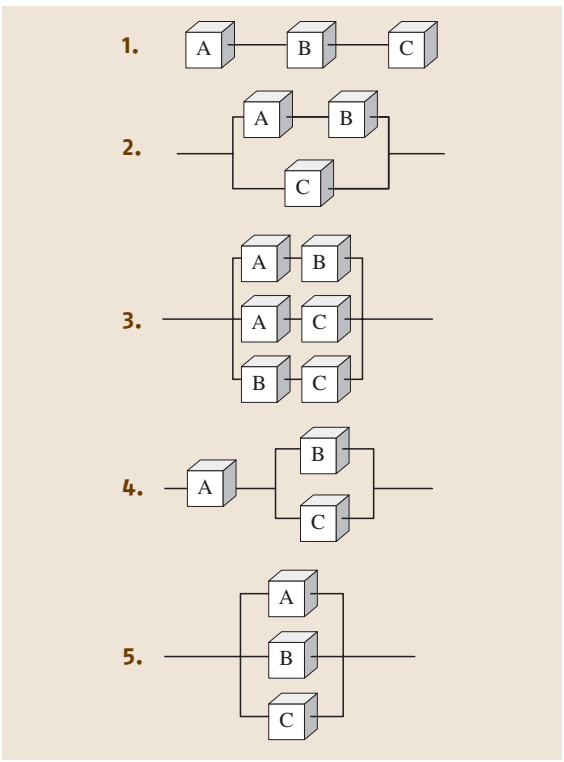


Fig. 2.3 Five unique systems of three components: (1) is series, (3) is 2-out-of-3 and (5) is parallel

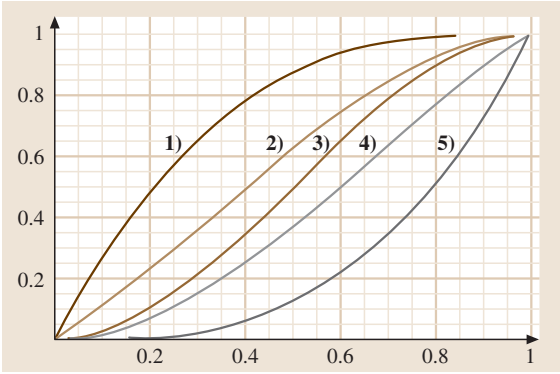


Fig. 2.4 System reliabilities of five system configurations in Fig. 2.3 from the series system (1) to the parallel system (5)

tests look to component lifetime data within a working system. For a k-out-of-n system, for example, the system lifetime represents an order statistic of the underlying distribution function. That is, if the ordered lifetimes form a set of independent and identically distributed components ($X_{1:n} \le X_{2:n} \le \cdots \le X_{n:n}$), then $X_{n-k+1:n}$ represents the k-out-of-n system lifetime. The density function for $X_{r:n}$ is

$$f_{r:n}(t) = r\binom{n}{r} F(t)^{r-1}\,[1-F(t)]^{n-r}\,f(t), t>0\,. \tag{2.19}$$

Kvam and *Samaniego* [2.32] derived the nonparametric maximum likelihood estimator for $F(t)$ based on a sample of k-out-of-n system data, and showed that the MLE $\hat{F}(t)$ is consistent. If the i-th system ($i = 1, \ldots, m$) observed is a k_i-out-of-n_i system, the likelihood can be represented as

$$L(F) = \prod_{i=1}^{m} f_{k_i:n_i}(t_i) \tag{2.20}$$

and numerical methods are employed to find $\hat{F}$. *Huang* [2.33] investigated the asymptotic properties of this MLE, and *Chen* [2.34] provides an ad hoc estimator that examines the effects of censoring.

Compared to individual component tests, observed system lifetimes can be either advantageous or disadvantageous. With an equal number of k-out-of-n systems at each $1 \le k \le n$, *Takahasi* and *Wakimoto* [2.35] showed that the estimate of MTTF is superior to that of an equal number of individual component tests. With an unbalanced set of system lifetimes, no such guarantee can be made. If only series systems are observed, *Kvam* and *Samaniego* [2.36] show how the uncertainty in $\hat{F}(t)$ is relatively small in the lower quantiles of F (where system failures are observed) but explodes in the upper quantiles.

2.4.2 Stochastic Dependence Between System Components

Almost all basic reliability theory is based on systems with independently operating components. For realistic modeling of complex systems, this assumption is often impractical; system components typically operate at a level related to the quality and operational state of the other system components.

External events that cause the simultaneous failure of component groups is a serious consideration in reliability analysis of power systems. This can be a crucial point in systems that rely on built-in component redundancy to achieve high target system reliability. Shock models, such as those introduced by *Marshall* and *Olkin* [2.37], can be employed to demonstrate how multiple component failures can occur. An extra failure process is added to the otherwise independent component failure processes, representing the simultaneous failure of one or more components, thus making the component lifetimes positively dependent. This is the basis for most dependent failure models in probabilistic risk assessment, including *common cause failure* models used in the nuclear industry (alpha-factor model, beta-factor model, binomial failure rate model). See Chapt. 8 of *Bedford* and *Cooke* [2.16] for discussion about how these models are used in risk assessment.

In dynamic systems, where system configurations and component reliability can change after an external event or a failure of one or more of the system components, the shock model approach cannot be applied effectively. In some applications, a load-share model applies.

Early applications of the load-share system models were investigated by *Daniels* [2.38] for studying the reliability of composite materials in the textile industry. Yarns and cables fail after the last fiber (or wire) in the bundle breaks, thus a bundle of fibers can be considered a parallel system subject to a constant tensile load. An individual fiber fails in time with an individual rate that depends on how the unbroken fibers within the bundle share the load of this stress. Depending on the physical properties of the fiber composite, this load sharing has different meanings in the failure model. Yarn bundles or untwisted cables tend to spread the stress load uniformly after individual failures which defines an *equal*

load-share rule, implying the existence of a constant system load that is distributed equally among the working components.

As expected, a load-sharing structure within a system can increase reliability (if the load distribution saves the system from failing automatically) but reliability inference is hampered even by the simplest model. *Kvam* and *Peña* [2.39] show how the efficiency of the load-share system, as a function of component dependence, varies between that of a series system (equivalent to sharing an infinite load) and a parallel system (equivalent to sharing zero load).

2.4.3 Logistics Systems

Numerous studies have examined fundamental problems in network reliability [2.40], system performance degradation and workload rerouting for telecommunication, power and transportation networks [2.41, 42]. In comparison, the literature on modeling logistics system reliability or performance degradation is scarce. Logistics systems that transport goods, energy (e.g., electricity and gas), water, sewage, money or information from origins to destinations are critical to every nation's economic prosperity. Unlike the hub in the typical Internet or telecommunication or network, where the messages are not mixed together, logistics distribution centers (DCs) tend to mix products from various sources for risk-pooling purposes [2.35]. Past studies [2.43] of road-network reliability mainly addressed connectivity and travel-time reliability. These developments have limited use in providing a first-cut analysis for system-level planning that involves robust logistics network design to meet reliability requirements or supply-chain cost and delivery-time evaluation for contract decisions [2.44].

Consider a logistics network consisting of many suppliers providing goods to several DCs, which support store operations to meet customer demands. The reliability of such a network can be evaluated in terms of the probability of delivering goods to stores in a prespecified time limit t_0. Traveling time in transport routes contains uncertainty, as does the processing time for products shipped through DCs. Random traveling time is a function of routing distances, road and traffic conditions and possible delays from seaport or security checkpoint inspections. Traveling distance depends on the configuration of logistics networks. Some retail chains use single-layer DCs, but others use multiple-layer DCs similar to airline hubs (e.g., regional DCs and global DCs) in aggregating various types of goods. Vehicle routing procedures typically involve trucks that carry similar products to several stores in an assigned region. Different products are consolidated in shipment for risk-pooling purposes and to more easily control delivery-time and store-docking operations.

When one DC cannot meet the demands from its regional stores (due to demand increase or the DC's limited capability), other DCs provide backup support to maintain the overall network's service reliability. Focusing on the operations between DCs and stores, *Ni* et al. [2.45] defined the following network reliability as a weighted sum of the individual reliabilities from each DC's operations:

$$r^*_{\text{system},k} = \left[\sum_{i=1, i\neq k}^{M} d_i P\left(T^*_{m,i} < t_0\right) + \sum_{i=1, i\neq k}^{M} p_i d_k P\left(T^*_{m,k,i} < t_0\right) \right] \Big/ \sum_{i=1}^{M} d_i \,, \tag{2.21}$$

where d_i is the demand aggregated at the i-th DC, $T^*_{m,i}$ is the motion time defined as the sum of traveling time from DC_i to its assigned stores (including material processing time at DC_i), p_i is the proportion of products rerouted from DC_k through DC_i due to the limited capability in DC_k and $T^*_{m,k,i}$ is the modified motion time including the rerouted traveling time.

For modeling the aggregated demand d_i and calculating routing distance, *Ni* et al. [2.45] proposed a multiscale approximation model to quantify demand patterns at spatially located clustered stores. Then, they evaluated product rerouting strategies for maintaining system service reliability, defined in (2.19). Based on the store locations of a major retail chain, several examples show the importance of designing a robust logistics network to limit service reliability degradation when a *contingency* (e.g., multiple DC failure) occurs in the network. Future work includes:

1. Modeling the low-probability but high-impact contingency in the DCs [2.45] and routes for calculating their relative importance to network reliability
2. Examining the tradeoff between the cost of adding more DCs and the improvement of service reliability
3. Resolving the *domino effect* when the added workload to DCs after a local DC failure causes further DC failures due to faulty predetermined rules of rerouting to maintain system reliability (e.g., the 2003 electricity blackout in the Northeastern region of the US).

2.4.4 Robust Reliability Design in the Supply Chain

Past studies of robust parameter design [2.46] focused on product quality issues and assumed that all the controllable variables are under single ownership. Recent outsourcing trends in automobile and electronic manufacturing processes motivate the presentation in this section. In an automobile manufacturing enterprize system, various parts suppliers have control of variables determining quality and reliability. Most of the automobile supply-chain systems assemble these parts into a subsystem and then move these systems to other locations owned by different partners for the final system-level assembly and testing. Every segment of the assembly operation controls a subset of variables leading to different levels of system reliability. Because the warranty policy is addressed to all of the part manufacturing and assembly processes in making the final product, it is important to extend the robust parameter design concept to the supply-chain-oriented manufacturing processes.

Supply-chain partners have their own operation objectives (e.g., maximize the profit of manufacturing parts to supply several automobile companies). Some of the objectives are aligned to manufacturing a specific type of product, but there are many potential situations with conflicting objectives. When there is no single ownership of all controllable variables in the manufacturing processes, negotiation is needed to resolve potential conflicts. Game theory [2.47] is commonly used in supply-chain contract decisions. Following the framework of *Chen* and *Lewis* [2.48], we can decompose the set of controllable variables into a few subsets owned by distinct partners and formulate the objectives of these partners. The supply-chain manager can define the product quality and reliability measures and build models to link them to the controllable and uncontrollable variables that are seen in robust parameter design.

Different negotiation situations (e.g., the final product assembly company has more bargaining power than other partners) will lead to distinct levels selected for the controllable variables (see *Charoensiriwath* and *Lu* [2.49] for examples in negotiations). As a result, the reliability of the final product can vary. Designing a supply-chain system that leads to the most reliable products (with minimum cost) presents an acute challenge, and warranty policies can be designed correspondingly. Because parts and subsystems are made by various partners, warranty responsibilities for certain parts are distributed among partners under the negotiated supply-chain contracts.

References

2.1 Z. W. Birnbaum, S. C. Saunders: A new family of life distributions, J. Appl. Probab. **6**, 319–327 (1969)

2.2 R. E. Barlow, F. Proschan: *Statisitcal Theory of Reliability and Life Testing* (Holt, Rinehart, Austin 1975)

2.3 J. I. Ansell, M. J. Phillips: *Practical Methods for Reliability Data Analysis* (Oxford Univ. Press, Oxford 1994)

2.4 L. Leemis: *Reliability: Probabilistic Models and Statistical Methods* (Prentice Hall, Englewood 1995)

2.5 P. A. Tobias, D. C. Trindade: *Applied Reliability* (CRC, Boca Raton 1995)

2.6 R. E. Barlow: *Engineering Reliability* (Society for Industrial and Applied Mathematics, Alexandria 1998)

2.7 P. Kales: *Reliability for Technology, Engineering and Management* (Prentice–Hall, Englewood 1998)

2.8 W. Q. Meeker, L. A. Escobar: *Statistical Methods for Reliability Data* (Wiley, New York 1998)

2.9 W. R. Blischke, D. N. P. Murthy: *Reliability Modeling, Prediction, and Optimization* (Wiley, New York 2000)

2.10 S. E. Rigdon, A. P. Basu: *Statistical Methods for the Reliability of Repairable Systems* (Wiley-Interscience, New York 2000)

2.11 V. E. Johnson, M. Hamada, H. Martz, S. Reese, A. Wilson: *Modern Reliability Analysis: A Bayesian Perspective* (Springer, Berlin Heidelberg New York 2005)

2.12 J. L. Bogdanoff, F. Kozin: *Probabilistic Models of Cumulative Damage* (Wiley, New York 1985)

2.13 K. Sobczyk, B. F. Spencer: *Random Fatigue From Data to Theory* (Academic, Boston 1992)

2.14 M. R. Lyu: *Handbook of Software Reliability Engineering* (McGraw Hill, New York 1996)

2.15 N. D. Singpurwalla, S. P. Wilson: *Statistical Methods in Software Engineering* (Springer, Heidelberg Berlin New York 1999)

2.16 T. Bedford, R. Cooke: *Probabilistic Risk Analysis: Foundations and Methods* (Cambridge Univ. Press, London 2001)

2.17 G. Baecher, J. Christian: *Reliability and Statistics in Geotechnical Engineering* (Wiley, New York 2003)

2.18 M. Ohring: *Reliability & Failure of Electronic Materials & Devices* (Academic, Boston 1998)

2.19 R. E. Melchers: *Structural Reliability Analysis and Prediction* (Wiley, New York 1999)

2.20 M. Shaked, J. G. Shanthikumar: *Stochastic Orders and their Applications* (Academic, Boston 1994)
2.21 W. Weibull: A statistical theory of the strength of material, Ingeniors Vetenskaps Akademiens Handligar, Stockholm **151**, 5–45 (1939)
2.22 H. A. David: *Order Statistics* (Wiley, New York 1981)
2.23 G. S. Mudholkar, D. K. Srivastava, G. D. Kollia: A generalization of the Weibull distribution with application to the analysis of survival data, J. Am. Stat. Assoc. **91**, 1575–1583 (1996)
2.24 A. Høyland, M. Rausand: *System Reliability Theory Models and Statistical Methods* (Wiley, New York 1994)
2.25 W. Q. Meeker: SPLIDA (S-PLUS Life Data Analysis) Version 6.5, http://www.public.iastate.edu/~splida/
2.26 S. J. Bae, P. H. Kvam: A nonlinear random coefficients model for degration testing, Technometrics **46**(4), 460–469 (2004)
2.27 S. J. Bae, P. H. Kvam: A change-point analysis for modeling incomplete burn-in for light displays, IEEE Trans. , to appear (2006)
2.28 A. C. Atkinson: *Plots, Transformations, and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*, Oxford Statistical Science Series (Oxford Univ. Press, Oxford 1992)
2.29 L. A. Waller, B. W. Turnbull: Probability Plotting with censored data, Am. Stat. **46**, 5–12 (1992)
2.30 A. Owen: *Empirical Likelihood* (Chapman and Hall, CRC, Boca Raton 2001)
2.31 B. Efron, R. Tibshirani: *An Introduction to the Bootstrap* (Chapman and Hall, CRC Press, Boca Raton 1993)
2.32 P. H. Kvam, F. J. Samaniego: Nonparametric maximum likelihood estimation based on ranked set samples, J. Am. Stat. Assoc. **89**, 526–537 (1994)
2.33 J. Huang: Asymptotic of the NPMLE of a distribution function based on ranked set samples, Ann. Stat. **25**(3), 1026–1049 (1997)
2.34 Z. Chen: Component reliability analysis of k-out-of-n systems with censored data, J. Stat. Plan. Inference. **116**, 305–316 (2003)
2.35 K. Takahasi, K. Wakimoto: On unbiased estimates of the population mean beased on samples stratified by meand of ordering, Ann. Inst. Stat. Math. **20**, 1–31 (1968)
2.36 P. H. Kvam, F. J. Samaniego: On estimating distribution functions using nomination samples, J. Am. Stat. Assoc. **88**, 1317–1322 (1993)
2.37 A. W. Marshall, I. Olkin: A multivariate exponential distribution, J. Am. Stat. Assoc. **62**, 30–44 (1967)
2.38 H. E. Daniels: The statistical theory of the strength of bundles of threads, Proc. R. Soc. Lond. A **183**, 405–435 (1945)
2.39 P. H. Kvam, E. Peña: Estimating load-sharing properties in a dynamic reliability system, J. Am. Stat. Assoc. **100**(469), 262–272 (2004)
2.40 M. O. Ball: Computing Network Reliability, Oper Res **27**, 823–838 (1979)
2.41 B. Sanso, F. Soumis: Communication and Transportation Network Reliability Using Routing Models, IEEE Trans. Reliability **40**, 29–37 (1991)
2.42 V. Shi: Evaluating the performability of tactical communications network, IEEE Trans. Vehicular Technol. **53**, 253–260 (2004)
2.43 A. Chen, H. Yang, H. Lo, W. H. Tang: A capacity related reliability for transportation networks, J. Adv. Transport. **33(2)**, 183–200 (1999)
2.44 S. Dandamudi, J.-C. Lu: Competition Driving Logistics Design with Continuous Approximation Methods, Technical Report can be seen in http://www.isye.gatech.edu/apps/research-papers/
2.45 W. Ni, J.-C. Lu, P. Kvam: Multi-scale Spatial Modeling for Logistics System Reliability, Technical Report can be seen in http://www.isye.gatech.edu/apps/research-papers/
2.46 C. F. J. Wu, M. Hamada: *Experiments: Planning, Analysis, and Parameter Design Optimization* (Wiley, New York 2000) p. 503
2.47 D. Fudenberg, J. Tirole: *Game Theory* (MIT Press, Cambridge 2000)
2.48 W. Chen, K. Lewis: Robust design approach for achieving flexibility in multidisciplinary design, AIAA J. **37(8)**, 982–989 (1999)
2.49 C. Charoensiriwath, J.-C. Lu: Competition Under Retail Price and Manufacturing Service, Technical Report can be seen in http://www.isye.gatech.edu/apps/research-papers/

3. Weibull Distributions and Their Applications

Weibull models are used to describe various types of observed failures of components and phenomena. They are widely used in reliability and survival analysis. In addition to the traditional two-parameter and three-parameter Weibull distributions in the reliability or statistics literature, many other Weibull-related distributions are available. The purpose of this chapter is to give a brief introduction to those models, with the emphasis on models that have the potential for further applications. After introducing the traditional Weibull distribution, some historical development and basic properties are presented. We also discuss estimation problems and hypothesis-testing issues, with the emphasis on graphical methods. Many extensions and generalizations of the basic Weibull distributions are then summarized. Various applications in the reliability context and some Weibull analysis software are also provided.

The Weibull distribution is one of the best-known lifetime distributions. It adequately describes observed failures of many different types of components and phenomena. Over the last three decades, numerous articles have been written on this distribution. *Hallinan* [3.1] gives an insightful review by presenting a number of historical facts, the many forms of this distribution as used by practitioners, and possible confusions and errors that arise due to this non-uniqueness. *Johnson* et al. [3.2] devote a comprehensive chapter to a systematic study of this distribution. More recently, *Murthy* et al. [3.3] presented a monograph that contains nearly every facet relating to the Weibull distribution and its extensions.

In Sect. 3.1, we first define the three-parameter Weibull distribution and then look at its historical development and relations to other distributions. Section 3.2 studies the properties of the Weibull distribution, in particular those relevant to reliability. A brief discussion on simulation of Weibull variates is also included. We consider estimation problems and hypothesis testing in Sect. 3.3. In particular, we emphasize the graphical methods as a tool for selection and parameter estimation of a Weibull model. We devote Sect. 3.4 to Weibull-derived models, which includes many extensions and generalizations. Finally, in Sect. 3.5, we outline various applications, especially those in the reliability context. Because of the vast literature, we are unable to refer to all the source authors and we apologize in advance for any omissions in this regard.

Symbols and Abbreviations.

T	Random variable
$F(t)$	CDF, cumulative distribution function
$f(t)$	PDF, probability density function
$h(t)$	Failure rate function (hazard rate)
α, β, τ	Parameters
WPP	Weibull probability plot

3.1 Three-Parameter Weibull Distribution

According to *Hallinan* [3.1] the Weibull distribution has appeared in five different forms. The two common forms of the distribution function are as follows:

$$F(t) = 1 - \exp\left[-\left(\frac{t-\tau}{\alpha}\right)^{\beta}\right], \; t \geq \tau \tag{3.1}$$

and

$$F(t) = 1 - \exp\left[\lambda(t-\tau)^{\beta}\right], \; t \geq \tau \, . \tag{3.2}$$

The parameters of the distribution are given by the set $\boldsymbol{\theta} = \{\alpha, \beta, \tau\}$ with $\alpha > 0$, $\beta > 0$ and $\tau \geq 0$; where α is a scale parameter, β is the shape parameter that determines the appearance or shape of the distribution and τ is the location parameter. The parameter λ combines both scale and shape features as $\lambda = \alpha^{-\beta}$.

Although one should use $F(t, \boldsymbol{\theta})$ instead of $F(t)$, where $\boldsymbol{\theta} = (\alpha, \beta, \tau)$ denotes the vector of parameters, for notational convenience we suppress the parameter and use $F(t)$ to denote $F(t, \boldsymbol{\theta})$ in this chapter. Also, we do not intend to give an exhaustive review. Rather, we confine our discussion to aspects that are of relevance to the context of reliability theory.

For $\tau = 0$, (3.1) and (3.2) become the two-parameter Weibull distribution with

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\alpha}\right)^{\beta}\right], t \geq 0 \tag{3.3}$$

and

$$F(t) = 1 - \exp\left(-\lambda t^{\beta}\right), \; t \geq \tau \, . \tag{3.4}$$

Murthy et al. [3.3] refer to this as the *standard* Weibull model, *Johnson* et al. [3.2] refer to a standard Weibull when $\alpha = 1$ (or $\lambda = 1$) in (3.3), (3.4).

3.1.1 Historical Development

The Weibull distribution is named after its originator, the Swedish physicist Waloddi Weibull, who in 1939 used it to model the distribution of the breaking strength of materials [3.4] and in 1951 for a wide range of other applications [3.5]. The distribution has been widely studied since its inception. It has been known that Weibull may not be the first to propose this distribution. The name Fréchet distribution is also sometimes used due to the fact that it was *Fréchet* [3.6] who first identified this distribution to be an extremal distribution (later shown to be one of the three possible solutions by *Fisher* and *Tippett* [3.7]). According to *Hallinan* [3.4], it was Weibull who suggested a scale parameter and a location parameter that made the distribution meaningful and useful.

3.1.2 Relations to Other Distributions

The Weibull distribution includes the exponential ($\beta = 1$) and the Rayleigh distribution ($\beta = 2$) as special cases. If X denotes the Weibull variable, then $-X$ has a type 3 extreme-value distribution [3.8, Chapt. 22]. A simple log transformation will transform the Weibull distribution into the Gumbel distribution (type 1 extreme-value distribution).

The Burr XII distribution, is given by

$$F(t) = 1 - \left[1 + \left(\frac{t}{\alpha}\right)\right]^{-k}, t \geq 0 \, ; k \, , \alpha > 0 \, . \tag{3.5}$$

Let $k = \alpha$; as $k \to \infty$, then the Burr distribution approaches the Weibull (see, for example, [3.9]).

3.2 Properties

3.2.1 Basic Properties

Density Function

The probability density functions (PDF) (Fig. 3.1) of (3.1) and (3.2) are

$$f(t) = \beta\alpha^{-\beta}(t-\tau)^{\beta-1} \exp\left[-\left(\frac{t-\tau}{\alpha}\right)^{\beta}\right], \; t \geq \tau \tag{3.6}$$

and

$$f(t) = \beta\lambda(t-\tau)^{\beta-1} \exp\left[-\lambda(t-\tau)^{\beta}\right], \; t \geq \tau \, . \tag{3.7}$$

Mode

It follows from (3.6) that the mode is at $t = \alpha((\beta-1)/\beta)^{1/\beta} + \tau$ for $\beta > 1$ and at τ for $0 < \beta \leq 1$.

Median

It follows from (3.3) that the median of the distribution is at $\alpha(\log 2)^{1/\beta} + \tau$.

Moments

Let T denote the random variable from the three-parameter Weibull distribution given by (3.1). Then the transformed variable $T' = (T - \tau)/\alpha$, is the stan-

dard form (in the sense of *Johnson* et al. [3.2]) with the density function given by

$$f(t) = \beta t^{\beta-1} \exp\left(-t^{\beta}\right) \ , x > 0 \ , \beta > 0 \ . \tag{3.8}$$

The moments of T (about zero) are easily obtained from the moments of T' which are given below:

$$\mu_r^{'} = E(T'^r) = \Gamma\left(\frac{r}{\beta}+1\right) \ , \tag{3.9}$$

from which we get

$$E(T') = \Gamma\left(\frac{1}{\beta}+1\right) \ , \tag{3.10}$$

$$\mathrm{Var}(T) = \Gamma\left(\frac{2}{\beta}+1\right) - \left[\Gamma\left(\frac{1}{\beta}+1\right)\right]^2 \ . \tag{3.11}$$

Skewness and Kurtosis

The distribution is positively skewed for small values of β. The skewness index $\sqrt{\beta_1}$ decreases and equals zero for $\beta = 3.6$ (approximately). Thus, for values of β in the vicinity of 3.6, the Weibull distribution is similar in shape to a normal distribution. The coefficient of kurtosis β_2 also decreases with β and then increases, β_2 has a minimum value of about 2.71 when $\beta = 3.35$ (approximately).

Order Statistics

Let $T_1, T_2, \cdots, T_n$ denote n independent and identically distributed three-parameter Weibull random variables with density function given in (3.6) and cumulative distribution function (CDF) in (3.1). Further, let $T_{(1)} \le T_{(2)} \le \cdots \le T_{(n)}$ denote the order statistics from a sample of n observations. The probability density function of $T_{(1)}$, is given by

$$\begin{aligned} f_{(1)}(t) &= n[1-F(t)]^{n-1} f(x) \\ &= \frac{n\beta}{\alpha}\left(\frac{t-\tau}{\alpha}\right)^{\beta-1} \mathrm{e}^{-n[(t-\tau)/\alpha]^{\beta}}, t \ge \tau \ . \end{aligned} \tag{3.12}$$

It is obvious from (3.12) that $T_{(1)}$ is also distributed as a Weibull random variable, except that α is replaced by $\alpha n^{-1/\beta}$.

The density function of $T_{(r)} (1 \le r \le n)$ is given by

$$\begin{aligned} f_{(r)}(t) = \frac{n!}{(r-1)!(n-r)!}&\left(1-\mathrm{e}^{-[(t-\tau)/\alpha]^{\beta}}\right)^{r-1} \\ &\mathrm{e}^{-[(t-\tau)/\alpha]^{\beta}(n-r+1)} \beta\alpha^{-\beta}(t-\tau)^{\beta-1}, \ t \ge 0 \ . \end{aligned} \tag{3.13}$$

It can be shown that

$$E\left[\left(T_{(r)}\right)^k\right] = \sum_{i=0}^{k} \tau^i \alpha^{k-i} \omega_{(r)}^{k-i} \ , \tag{3.14}$$

where

$$\begin{aligned} \omega_{(r)}^{k} = &\frac{n!}{(r-1)!(n-r)!}\Gamma\left(1+\frac{k}{\beta}\right) \\ &\times \sum_{i=0}^{r-1} \frac{(-1)^r \binom{r-1}{i}}{(n-r+i+1)^{1+(k/\beta)}} \ . \end{aligned}$$

3.2.2 Properties Related to Reliability

In this section, we consider only the first form of the Weibull distribution.

The survival function of the Weibull distribution is

$$\bar{F}(t) = 1 - F(t) = \exp\left[-\left(\frac{t-\tau}{\alpha}\right)^{\beta}\right], \ t \ge \tau \ . \tag{3.15}$$

Note that at $t = \tau + \alpha$, $\bar{F}(\tau+\alpha) = 1 - \mathrm{e}^{-1} \approx 0.3679$.

Failure Rate

The failure rate function (also known as the hazard rate) for the three-parameter Weibull is

$$h(t) = \frac{f(t)}{\bar{F}(t)} = \frac{\beta}{\alpha}\left(\frac{t-\tau}{\alpha}\right)^{\beta-1} \ . \tag{3.16}$$

For the two-parameter case, it is given by

$$h(t) = \frac{f(t)}{\bar{F}(t)} = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \ . \tag{3.17}$$

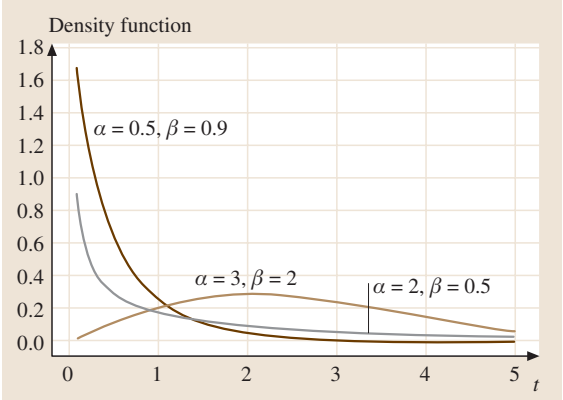


Fig. 3.1 Two-parameter Weibull density plots

It is obvious that $h(t)$ is a decreasing function when $\beta < 1$, constant when $\beta = 1$ (the exponential case), and an increasing function when $\beta > 1$. Because of the behaviour of the failure rate function, the Weibull distribution often becomes suitable when the conditions for *strict randomness* of the exponential distribution are not satisfied, with the shape parameter β having a value depending upon the fundamental nature being considered.

In some way, having a failure rate function of monotonic shape has limitations in reliability applications. For this reason, several generalized or modified Weibull distributions have been proposed (see Sect. 3.4 for more details). Figure 3.2 shows plots of the failure rate functions for some selected parameters values.

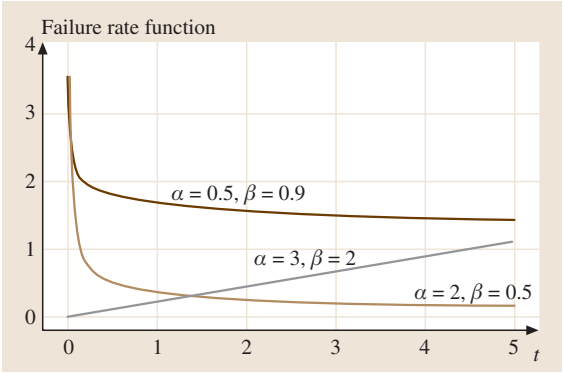


Fig. 3.2 Two-parameter Weibull failure rate functions

Mean Residual Life

The mean residual life (MRL) of a lifetime random variable T is defined as

$$\mu(t) = E(T - t | T > t) = \frac{\int_t^\infty \bar{F}(x)\,\mathrm{d}x}{\bar{F}(t)} \,. \tag{3.18}$$

For the Weibull distribution, the MRL is complicated except for the two special cases $\beta = 1$ (exponential) and $\beta = 2$ (Rayleigh distribution). Assuming the location parameter $\tau = 0$, the MRL of the Rayleigh distribution with scale parameter $\alpha = \sqrt{2}\sigma$ is

$$\mu(t) = \sqrt{2\pi}\sigma\,[1 - \Phi(t/\sigma)]\,\mathrm{e}^{\frac{1}{2\sigma^2}t^2} \,, \quad t > 0 \,, \tag{3.19}$$

where $\Phi(t/\sigma)$ denotes the distribution function of the standard normal variable.

Relative Ageing of Two Two-Parameter Weibull Distributions

Suppose we have two Weibull random variables X and Y with distribution functions $F(x)$ and $G(y)$, respectively, given by

$$\begin{aligned} F(x) &= 1 - \exp\left[-(x/\alpha_2)^{\beta_2}\right] \,, \\ G(y) &= 1 - \exp\left[-(y/\alpha_1)^{\beta_1}\right] \,. \end{aligned} \tag{3.20}$$

We say that X ages faster than Y if the ratio of the failure rate of X over the failure rate of Y is an increasing function of t. This ratio is given by

$$\frac{\alpha_1\beta_2}{\alpha_2\beta_1} \times \frac{\alpha_1^{\beta_2 - 1}}{\alpha_2^{\beta_1 - 1}} t^{\beta_2 - \beta_1} \tag{3.21}$$

which is an increasing function of t if $\beta_2 > \beta_1$.

Suppose $E(X) = E(Y)$, i. e., $\alpha_2\Gamma(1 + 1/\beta_2) = \alpha_1\Gamma(1 + 1/\beta_1)$. *Lai* and *Xie* [3.10] show that $\mathrm{Var}(X) \le \mathrm{Var}(Y)$.

Effectiveness of Parallel Redundancy – The Weibull Case

Denote the lifetime of one component by T and the lifetime of a parallel system of two such independent components by T_p. The effectiveness of parallel redundancy of the component is defined by [3.11]

$$e_\mathrm{p} = [E(T_\mathrm{p}) - E(T)]/E(T) \,. \tag{3.22}$$

Suppose that T is a two-parameter Weibull distribution with shape parameter β and scale parameter α. *Xie* and *Lai* [3.11] show that

$$e_\mathrm{p} = 1 - 2^{-1/\beta} \,, \tag{3.23}$$

which decreases as β increases. Thus, a parallel redundancy is more effective for $\beta < 1$.

3.2.3 Simulation

The two-parameter Weibull distribution with parameters β and λ has the probability density function

$$f(x) = \lambda\beta x^{\beta - 1}\,\mathrm{e}^{-\lambda x^\beta} \,. \tag{3.24}$$

The simple inverse-probability integral-transform method applied to the standard Weibull distribution ($\lambda = 1$) is quite efficient. The formula is simply

$$x = (-\log u)^{1/\beta} \,,$$

where u is a uniform number between 0 and 1. Of course, an acceptance/rejection method could also be used to avoid the evaluation of the logarithm in the distribution function. The standard Weibull variate is then scaled by $\lambda^{-(1/\beta)}$ to obtain a two-parameter Weibull observation with density (3.24).

3.3 Modeling Failure Data

Failure data can be classified into two types: complete and incomplete (censored). For complete data, the actual realized values are known for each observation in the data set, whereas for censored data, the actual realized values are not known for some or all of the observations. Further, there are at least two types of censoring, details of which may be found *Lawless* [3.12] and *Nelson* [3.13].

In their introduction, *Murthy* et al. [3.3] point out that the two different approaches to building mathematical models are as follows.

1. *Theory-based modeling*: here, the modeling is based on the established theories for component failures. This kind of model is also called a *physics based model* or *white-box model*.
2. *Empirical modeling*: here, the data available forms the basis for the model building. This kind of model is also called a *data-dependent model* or *black-box model*.

In empirical modeling, the type of mathematical formulations needed for modeling is dictated by a preliminary analysis of data available. If the analysis indicates that there is a high degree of variability, then one needs to use models that can capture this variability. This requires probabilistic and stochastic models to model a given data set.

The process of black box modeling involves the following steps:

- Step 1 – Model selection,
- Step 2 – Parameter estimation,
- Step 3 – Model validation.

To select a model out of many possible models, one requires a good understanding of their different properties. This often is a trial-and-error process. *Liao* and *Shimokawa* [3.14] consider goodness-of-fit testing as a key procedure for selecting the statistical distribution that best fits the observed data. To execute the remaining two steps, we need various tools and techniques.

A large number of Weibull-related models have been derived in the literature. In Sect. 3.4, various such models are grouped into several categories. Selecting an appropriate model from the family of Weibull-related distributions can often be based on some of the probability plots to be discussed below. Moreover, they can provide crude estimates of model parameters. There are many different statistical tests for validating a model and we postpone this discussion until the next section.

3.3.1 Probability Plots

Weibull Probability Plot

Weibull probability plots (WPP) can be constructed in several ways [3.15]. In the early 1970s a special paper was developed for plotting the data in the form of $F(t)$ versus t on a graph paper with a log–log scale on the vertical axis and a log scale on the horizontal axis. A WPP plotting of data involves computing the empirical distribution function, which can be estimated in different ways; the two standard methods are:

- $\hat{F}(t_i) = i/(n+1)$, the "mean rank" estimator, and
- $\hat{F}(t_i) = (i-0.5)/n$, the "median rank" estimator.

Here, the data consists of successive failure times t_i, $t_1 < t_2 < \ldots t_n$. For censored data (right-censored or interval), the approach to obtain the empirical distribution functions needs to be modified; see, for example, *Nelson* [3.13].

These days, most computer reliability software packages contain programs to produce these plots automatically given a data set. A well-known statistical package MINITAB provides a Weibull probability plot under the Graph menu ≫ Probability Plot.

We may use an ordinary graph paper or spreadsheet software with unit scale for plotting. Taking logarithms twice of both sides of each of the CDF in (3.3) yields

$$\log[-\log \bar{F}(t)] = \beta \log(t-\tau) - \beta \log \alpha \,. \tag{3.25}$$

Let $y = \log\left[-\log \bar{F}(t)\right]$ and $x = \log(t-\tau)$. Then we have

$$y = \beta x - \beta \log \alpha \,. \tag{3.26}$$

The plot is now on a linear scale.

Weibull Hazard Plot

The hazard plot is analogous to the probability plot, the principal difference being that the observations are

plotted against the cumulated hazard (failure) rate rather than the cumulated probability value. Moreover, this is designed for censored data.

Let $H(t)$ denote the cumulative hazard rate, then $\bar{F}(t) = \exp[-H(t)]$, so

$$H(t) = -\log \bar{F}(t) = \left(\frac{t-\tau}{\alpha}\right)^{\beta}, \tag{3.27}$$

$$H(t)^{1/\beta}/\alpha = (t-\tau)\,. \tag{3.28}$$

Let $y = \log(t-\tau)$ and $x = \log H(t)$, then we have

$$y = \log\alpha + \frac{1}{\beta}x\,. \tag{3.29}$$

Rank the n survival times (including the censoreddata) in ascending order and let K denote the reverse ranking order of the survival time, i. e., $K = n$ for the smallest survival time and $K = 1$ for the largest survival time. The hazard is estimated from $100/K$ (a missing value symbol is entered at a censored failure time). The cumulative hazard is obtained by cumulating the hazards. The Weibull hazard plot is simply the plot arising from (3.29). See *Nelson* [3.16] for further details.

3.3.2 Estimation and Hypothesis Testing

Parameter estimation is the second step of our modeling process. We consider both graphical and statistical methods for the three-parameter Weibull distribution.

Graphical Methods

Graphical plots are designed not just to assess if the data follows a Weibull population, they are also useful for estimating Weibull parameters (at least for initial estimates). These can be obtained from the smooth fit to a WPP plot of data and involve exploiting properties such as asymptotes, intersection points, slope, or points of inflection.

Estimation of Location Parameter. The determination of a suitable location parameter τ is not a simple task. If a data set graphs as a straight line on a WPP, then the data set is indeed adequately described by a Weibull distribution. However, an incorrect selection of τ will yield a curved plot of data that is in fact Weibull-distributed.

If the graph of a set of data appears concave upward, then the plot can be straightened by decreasing the value of τ. Conversely, if the data are concave downward, then the plot can be straightened by increasing the value of τ. The value of τ cannot be chosen larger than the smallest failure-time value, say $T_{(1)}$, in the sample. It is generally suggested to set the initial value of $\tau = T_{(1)}$ and then adjust this estimate to achieve a straight line. We also note that it is common for τ to be 0.

Estimation of the Shape Parameter. It follows from (3.26) that the shape parameter can be estimated from the slope of the WPP. However, care needs need to be given because possible confusion may arise due to several possible forms of the Weibull distribution.

Estimation of the Scale Parameter. To estimate the scale parameter, we first observe from the standard Weibull that, when t is at the 63.2 percentile, $F(t) = 1 - e^{-1}$, i. e., $\bar{F}(t) = e^{-1}$. Or equivalently, for $t = \tau + \alpha$, $\bar{F}(\tau+\alpha) = e^{-1}$ (so $\tau+\alpha$ is the 63.2 percentile). It now follows from (3.26) that $\log(t-\tau) = \log\alpha$ so that $\hat{\alpha} = t - \hat{\tau}$, where t is the 63.2 percentile. Hence the scale parameter is simply estimated from the x-intercept of the plot (3.26): $\hat{\alpha} = e^{(x\text{-intercept})}$.

Under the probability plot option, MINITAB not only gives a WPP, but also provides an estimate for the scale and shape parameters α and β.

Figure 3.3 illustrates how a WPP plot appears in a MINITAB graph with $\hat{\alpha} = 9.062$ and $\hat{\beta} = 1.661$.

Statistical Methods of Estimation

There are several statistical methods for estimating the model parameters. These include the method of moment, the method of percentile, the method of maximum likelihood, and the Bayesian method. The estimates can be

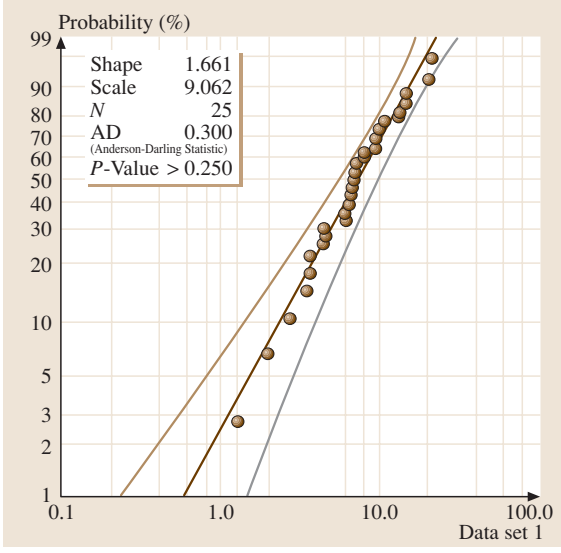


Fig. 3.3 A typical Weibull probability plot

either point estimates or interval estimates. The details of these methods can be found, for example, in *Kalbfleisch* and *Prentice* [3.17].

We note that these methods may not be appropriate for small data sets. Hence, the estimates obtained from the WPP plot may be taken as the final estimates when the data size is small.

Moment Estimation. By equating the first three moments of the Weibull distribution to the first three sample moments and solving, it is possible to find the moment estimators of α, β and τ. Now, the first moment ratio $\sqrt{\beta_1}$ depends only on the shape parameter β, and hence once $\sqrt{\beta_1}$ has been estimated from the sample coefficient of skewness, $\hat{\beta}$ can be determined numerically. Using $\hat{\beta}$, $\hat{\alpha}$ can be determined from the standard deviation, and lastly $\hat{\tau}$ is determined from the sample mean. If τ is known, then β can be estimated from the ratio (standard deviation/(mean-τ).

For modified moment estimation, see *Cohen* et al. [3.18].

Maximum-Likelihood Method. Maximum-likelihood methods for the three-parameter Weibull distribution have been reviewed by *Zanakis* and *Kyparisis* [3.19]; see also *Johnson* et al. [3.2, Chapt. 21] for other details.

The most common situation is when the location parameter τ is known. Without loss of generality, we may assume that $\tau = 0$ so that the model becomes the two-parameter Weibull distribution.

The maximum-likelihood estimators, $\hat{\beta}$ and $\hat{\alpha}$ of β and α, respectively, satisfy the equations

$$\hat{\alpha} = \left(\frac{1}{n} \sum_{i=1}^{n} X_i^{\hat{\beta}} \right)^{1/\hat{\beta}} \tag{3.30}$$

and

$$\hat{\beta} = \left[\left(\sum_{i=1}^{n} X_i^{\hat{\beta}} \log X_i \right) \left(\sum_{i=1}^{n} X_i^{\hat{\beta}} \right)^{-1} - \frac{1}{n} \sum_{i=1}^{n} \log X_i \right]^{-1} , \tag{3.31}$$

where X_i, $i = 1, 2, \ldots n$ are n independent observations from the two-parameter Weibull distribution. If $\tau \neq 0$, then each X_i is replaced by $X_i - \tau$ in the above equations. The value $\hat{\beta}$ needs to be solved from (3.31) and then (3.30) is used to obtain $\hat{\alpha}$.

Suppose the location parameter τ is also unknown, then the maximum-likelihood estimates $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\tau}$ satisfy the following equations

$$\hat{\alpha} = \left[\frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\tau})^{\hat{\beta}} \right]^{1/\hat{\beta}} \tag{3.32}$$

$$\hat{\beta} = \left\{ \left[\sum_{i=1}^{n} (X_i - \hat{\tau})^{\hat{\beta}} \log(X_i - \hat{\tau}) \right] \times \left[\sum_{i=1}^{n} (X - \hat{\tau})^{\hat{\beta}} \right]^{-1} - \frac{1}{n} \sum_{i=1}^{n} \log(X_i - \hat{\tau}) \right\}^{-1} \tag{3.33}$$

and

$$(\hat{\beta} - 1) \sum_{i=1}^{n} (X_i - \hat{\tau})^{-1} = \hat{\beta} \hat{\alpha}^{-\hat{\beta}} \sum_{i=1}^{n} (X_i - \hat{\tau})^{\hat{\beta}-1} . \tag{3.34}$$

If the value $\hat{\tau}$ satisfying (3.32)–(3.34) is larger than $X_1 = \min X_i$, then it is the maximum-likelihood estimate of τ. Otherwise, we set that $\hat{\tau} = X_1$ and then (3.32) and (3.33) must be solved for $\hat{\alpha}$ and $\hat{\beta}$.

The method is suitable for large data sets because of its asymptotic properties. Modifications of these estimates can be found, for example, in *Johnson* et al. [3.2].

The reader may consult Chapt. 21 of *Johnson* et al. [3.2] and Chapt. 4 of *Murthy* et al. [3.3] for the following methods of estimation,

- Best linear unbiased estimation,
- Percentile estimator,
- Bayesian estimator,
- Interval estimation,
- Minimum quantile distance estimation.

For the two-parameter Weibull distribution (i. e., $\tau = 0$), methods of inference can be obtained via the type 1 extreme-value distribution.

3.3.3 Hypothesis Testing

Goodness-of-Fit Tests for the Weibull Distribution

There are other general goodness-of-fit tests based on the empirical distribution functions derived. For example, the chi-square test, Kolmogorov–Smirnov test, Cramer–von Mises test and Anderson–Darling test, and so on, could also be used to test the goodness of fit of the Weibull model.

Generally, a goodness-of-fit test for Weibull distribution can be described as: H_0: the population follows a Weibull model, versus H_1: the Weibull model is not suitable.

MINITAB under probability plot gives the Anderson–Darling statistic as well as the confidence bands.

Lawless [3.12] summarizes the goodness-of-fit tests for the Weibull or extreme-value distribution. Four tests (the likelihood ratio test as a sub-model of the gamma distribution, Mann–Scheuer–Fertig test, Tiku test and Cramer–von Mises test) are discussed in detail. *Dodson* [3.20] also includes and compares some goodness-of-fit tests for Weibull distributions in Chapt. 4 of his book. *Liao* and *Shimokawa* [3.14] construct a new goodness-of-fit test for the two-parameter Weibull and its power is compared with other traditional goodness-of-fit tests.

Hypothesis Testing

Hypothesis testing of Weibull parameters may be performed by the likelihood ratio, score and Wald tests. It is well known that all these tests are asymptotically optimal.

For testing Weibull versus the exponential distribution, we test $\alpha = 1$ (or $\lambda = 1$ of the second form) versus $\alpha \neq 0$.

3.4 Weibull-Derived Models

There are many extensions, generalizations and modifications to the Weibull distribution. They arise out of the need to model features of empirical data sets that cannot be adequately described by a three-parameter Weibull model: for example, the monotonic property of the Weibull, which is unable to capture the behaviour of a data set that has a bathtub-shape failure rate. *Xie* et al. [3.21] review several Weibull-related distributions that exhibit bathtub-shaped failure rates. Plots of mean residual life from several of these Weibull-derived models are given in *Lai* et al. [3.22]. For simplicity, we simply refer to these Weibull-related models as Weibull models.

3.4.1 Taxonomy for Weibull Models

According to *Murthy* et al. [3.3], the taxonomy for Weibull models involves seven types, each of which divided into several sub-types. These models can be grouped into three groups:

1. Univariate models (types I–V),
2. Multivariate models (type VI),
3. Stochastic process models (type VII).

In this chapter we confine our discussion to univariate and stochastic process models, while *Murthy* et al. [3.23] looks at bivariate models (a special case of multivariate models). In this chapter, only some selective models will be included. We refer our readers to the book by *Murthy* et al. [3.3] for more details.

3.4.2 Univariate Models

The starting point is the two-parameter Weibull model with distribution function $F(t)$. Let $G(t)$ denote the derived Weibull model. T is a random variable from $F(t)$ and Z is a random variable from $G(t)$.

Type I Models (Transformation of Weibull Variable)

Here Z and T are related by a transformation. The transformation can be either linear or nonlinear. These include the following:

Reflected Weibull Distribution.

$$G(t) = \exp\left[-\left(\frac{\tau - t}{\alpha}\right)^{\beta}\right], -\infty < t < \tau\,. \quad (3.35)$$

This is also known as a type 3 extreme-value distribution (see *Johnson* et al. [3.8], Chapt. 22).

Double Weibull Distribution.

$$g(t) = \beta(1/2)\,|t|^{(\beta-1)} \exp(-|t|^{\beta})\,, -\infty < t < \infty\,. \quad (3.36)$$

This is an obvious extension to include the negative real line as its support.

Log Weibull Distribution. The distribution is derived from the logarithmic transformation of the two-parameter Weibull distribution. The distribution function is

$$G(t) = 1 - \exp\left[-\exp\left(\frac{t - \tau}{\alpha}\right)\right], -\infty < t < \infty\,. \quad (3.37)$$

This is also known as a type 1 extreme-value distribution or Gumbel distribution. In fact, it is the most commonly referred to in discussions of extreme-value distributions

(see [3.8], Chapt. 22). The density function is

$$g(t) = \frac{1}{\alpha} \exp\left(\frac{t-\tau}{\alpha}\right) \exp\left[-\exp\left(\frac{t-\tau}{\alpha}\right)\right] ,$$

and the failure rate function is quite simple and given by

$$h(t) = \frac{1}{\alpha} \exp\left(\frac{t-\tau}{\alpha}\right) .$$

Inverse (or Reverse) Weibull Model. Let $Z = \alpha^2/T$, where T has a two-parameter Weibull distribution, then the distribution function of T is

$$G(t) = \exp[-(t/\alpha)^{-\beta}] , t \geq 0 . \tag{3.38}$$

This is also known as a type 2 extreme-value distribution or Fréchet distribution (see [3.8], Chapt. 22). The failure rate function is as follows:

$$h(t) = \frac{\beta\alpha^{\beta} t^{-\beta-1} \mathrm{e}^{-(\alpha/t)^{\beta}}}{1 - \mathrm{e}^{-(\alpha/t)^{\beta}}} .$$

Type II Models (Modification/Generalization of the Weibull Distribution)

Here $G(t)$ is related to $F(t)$ by some relationship.

Extended Weibull Distribution [3.24].

$$G(t) = 1 - \frac{\nu \mathrm{e}^{-(\lambda t)^{\beta}}}{1 - (1-\nu)\mathrm{e}^{-(\lambda t)^{\beta}}} . \tag{3.39}$$

The failure rate function is

$$h(t) = \frac{\nu\beta(\lambda t)^{\beta-1}}{\left[1 - (1-\nu)\mathrm{e}^{-(\lambda t)^{\beta}}\right]} .$$

The failure rate function is increasing if $\nu \geq 1$, $\beta \geq 1$ and decreasing if $\nu \leq 1$, $\beta \leq 1$.

Modified Weibull Distribution [3.25].

$$G(t) = 1 - \exp(-\lambda t^{\beta} \mathrm{e}^{\nu t}) , t \geq 0 \tag{3.40}$$

with

$$h(t) = \lambda(\beta + \nu t) t^{\beta-1} \exp(\nu t) . \tag{3.41}$$

For $\nu = 0$, this reduces to a Weibull distribution. For $0 < \beta < 1$, $h(t)$ is initially decreasing and then increasing in t, implying that the failure rate function has a bathtub shape. When $\beta > 1$, $h(t)$ is increasing in t.

Exponentiated Weibull Distribution [3.26].

$$G(t) = [F(t)]^{\nu} = \{1 - \exp[-(t/\alpha)^{\beta}]\}^{\nu} , t \geq 0 . \tag{3.42}$$

The density function is

$$g(t) = \frac{\beta\nu}{\alpha^{\beta}} t^{\beta-1} \mathrm{e}^{-(t/\alpha)^{\beta}} \left\{1 - \mathrm{e}^{-(t/\alpha)^{\beta}}\right\}^{\nu-1} .$$

For an appropriate choice of the parameter set, it will give rise to a bathtub-shaped failure function. *Jiang* and *Murthy* [3.27] use a graphical approach to study this distribution. The failure rate function is given by

$$h(t) = \frac{\beta\nu}{\alpha^{\beta}} t^{\beta-1} \mathrm{e}^{-(t/\alpha)^{\beta}} \frac{\left(1 - \mathrm{e}^{-(t/\alpha)^{\beta}}\right)^{\nu-1}}{\left[1 - (1 - \mathrm{e}^{-(t/\alpha)^{\beta}})^{\nu}\right]} .$$

Four-Parameter Weibull Distribution [3.28].

$$G(t) = 1 - \exp\left[-\lambda\left(\frac{t-a}{b-t}\right)^{\beta}\right] ,$$
$$0 \leq a \leq t \leq b < \infty . \tag{3.43}$$

Doubly Truncated Weibull Distribution.

$$G(t) = \frac{F(t) - F(a)}{F(b) - F(a)} ,$$
$$0 < a \leq t \leq b < \infty . \tag{3.44}$$

Modified Weibull Extension [3.29].

$$G(t) = 1 - \exp\left[-\lambda\alpha\left(\mathrm{e}^{(t/\alpha)^{\beta}} - 1\right)\right] ,$$
$$t \geq 0, \alpha, \beta, \lambda > 0 . \tag{3.45}$$

This is known as the generalized exponential power model originally studied by *Smith* and *Bain* [3.30]. The case of $\alpha = 1$ is also studied in *Chen* [3.31]. The failure rate function is

$$h(t) = \lambda\beta(t/\alpha)^{\beta-1} \exp\left[(t/\alpha)^{\beta}\right] . \tag{3.46}$$

It approaches to a two-parameter Weibull distribution when $\lambda \to \infty$ with α in such a manner that $\alpha^{\beta-1}/\lambda$ is held constant.

Type III Models (Models Involving Two or More Distributions)

These are univariate models derived from two or more distributions with at least one being either the standard Weibull model or a distribution derived from it.

n-Fold Mixture Model.

$$G(t) = \sum_{i=1}^{n} p_i F_i(t)\,,\ p_i \geq 0\,,\ \sum_{i=1}^{n} p_i = 1\,. \qquad (3.47)$$

Jiang and *Murthy* [3.32, 33] categorize the possible shapes of the failure rate function for a mixture of any two Weibull distributions in terms of five parameters. *Gurland* and *Sethurama* [3.34] also consider a mixture of the Weibull distribution with failure rate $\lambda\beta t^{\beta-1}$ and the exponential distribution with failure rate λ_1. For $\beta > 1$, the Weibull distribution is IFR (Increasing failure rate). They found that the resulting mixture distribution is ultimately DFR (Decreasing failure rate). In fact, the mixture has a failure rate with an upside-down bathtub shape.

n-Fold Competing Risk Model.

$$G(t) = 1 - \prod_{i=1}^{n} [1 - F_i(t)]\,. \qquad (3.48)$$

Jiang and *Murthy* [3.35] also give a parametric study of a competing risk model involving two Weibull distributions.

n-Fold Multiplicative Model (Complementary Risk Model).

$$G(t) = \prod_{i=1}^{n} F_i(t)\,. \qquad (3.49)$$

The multiplicative model involving two Weibull distribution is considered in *Jiang* and *Murthy* [3.36].

n-Fold Sectional Model.

$$G(t) = \begin{cases} k_1 F_1(t)\,, & 0 \leq t \leq t_1\,, \\ 1 - k_2 \bar{F}_2(t)\,, & t_1 < t \leq t_2\,, \\ \cdots & \\ 1 - k_n \bar{F}_n(t)\,, & t > t_{n-1}\,, \end{cases} \qquad (3.50)$$

where the sub-populations $F_i(t)$ are two- or three-parameter Weibull distributions and the t_is (called partition points) are an increasing sequence. *Jiang* and *Murthy* [3.37] and *Jiang* et al. [3.38] consider the case $n = 2$ in details.

Type IV Models (Weibull Models with Varying Parameters)

For models belonging to this group, the parameters of the model are either functions of the independent variable (t) or some other variables (such as the stress level s, etc.), or are random variables.

Arrhenius Weibull Model.

$$\alpha(S) = \exp(\gamma_0 + \gamma_1 S)\,. \qquad (3.51)$$

Power Weibull Model.

$$\alpha(S) = \frac{e^{\gamma_0}}{S^{\gamma_1}}\,. \qquad (3.52)$$

These types of models have been used extensively in accelerated life testing [3.39] in reliability theory. As a result, they are referred to as *accelerated failure models*.

Weibull Proportional Hazard Models.

$$h(t) = \psi(S) h_0(t)\,, \qquad (3.53)$$

where $h_0(t)$ is called the baseline hazard for a two-parameter Weibull distribution. The only restriction on the scalar function $\psi(S)$ is that it be positive. Many different forms for $\psi(S)$ have been proposed. One such is the following:

$$\psi(S) = \exp\left(b_0 + \sum_{i=1}^{k} b_i s_i\right)\,. \qquad (3.54)$$

For more on such models, see *Cox* and *Oakes* [3.39] and *Kalbfleisch* and *Prentice* [3.17].

Type V Models (Discrete Weibull Models)

Here T can only assume non-negative integer values and this defines the support for $F(t)$.

Model 1 [3.40].

$$F(t) = \begin{cases} 1 - q^{t^\beta} & t = 0, 1, 2, 3 \cdots\,, \\ 0 & t < 0\,. \end{cases} \qquad (3.55)$$

Model 2 [3.41]. The cumulative hazard function is given by

$$H(t) = \begin{cases} c t^{\beta-1} & t = 1, 2, \cdots, m \\ 0 & t < 0\,, \end{cases} \qquad (3.56)$$

where m is given by

$$m = \begin{cases} \text{int}\left(c^{-[1/(\beta-1)]}\right) & \text{if} \quad \beta > 1\,, \\ \infty & \text{if} \quad \beta \leq 1\,, \end{cases} \qquad (3.57)$$

and int(•) represents the integer part of the quantity inside the brackets.

Model 3 [3.42].

$$F(t) = 1 - \exp\left[-\sum_{i=1}^{t} r(i)\right] = 1 - \exp\left[-\sum_{i=1}^{t} ci^{\beta-1}\right], \quad t = 0, 1, 2, \cdots . \tag{3.58}$$

3.4.3 Type VI Models (Stochastic Point Process Models)

These are stochastic point process models with links to the standard Weibull model.

Power Law Process (*Bassin* [3.43])

This is a point process model with the intensity function given by

$$\lambda(t) = \left(\frac{\beta t^{\beta-1}}{\alpha^{\beta}}\right) . \tag{3.59}$$

This model has been called by many different names: *power law process*; *Rasch–Weibull process*; *Weibull intensity function*); *Weibull–Poisson process* and *Weibull process*. We note that the inter-event times do not have Weibull distributions. For further discussion on the Weibull process, see for example, *Bain* and *Engelhart* [3.44, Chapt. 9].

Proportional Intensity Model (*Cox* and *Oaks* [3.45])

The intensity function is given by

$$\lambda(t; S) = \lambda_0(t)\psi(S) , t \geq 0 , \tag{3.60}$$

where $\lambda_0(t)$ is of the form given above and $\psi(S)$ is a function of the explanatory variables S. The only restriction on $\psi(S)$ is that $\psi(S) > 0$.

Ordinary Renewal Process (*Yannaros* [3.46])

Here the point process is a renewal process with the time between events being independent and identically distributed with the distribution function given by the standard Weibull distribution.

Modified Renewal Process

Here the distribution function for the time to first event, $F_0(t)$, is different from that for the subsequent inter-event times, which are identical and independent random variables with distribution function $F(\cdot)$. Note that $F_0(t)$ and/or $F(t)$ are standard Weibull distributions. When $F_0(t) = F(t)$, this reduces to the ordinary renewal process.

3.5 Empirical Modeling of Data

We have seen a large number of Weibull-related models which we simply refer as Weibull models. They exhibit a wide range of shapes for the density and failure rate functions, which make them suitable for modeling complex failure-data sets.

Recall, we mention that empirical modeling usually involves three steps: model selection, estimation of model parameters and model validation. In the context of Weibull models, a selection procedure may be based on WPP plots. This is possible due to the availability of WPP or generalized WPP plots for all the Weibull models of types I–III. Of course, the shape of the density and failure rate functions will also be valuable in the selection step. An added advantage of the WPP plots is that they provide crude estimates of model parameters; these serve as a starting point for steps 2 and 3.

It has been suggested that an alternative method to estimate model parameters is through a least-squares fit. Basically speaking, this involves selecting the parameters the parameters to minimize a function given

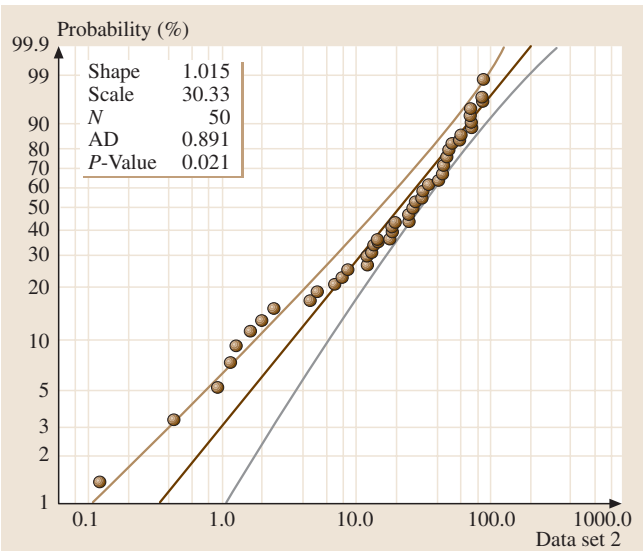


Fig. 3.4 Probability plot of data set from Table 3.1

Table 3.1 Data set of failure test (data set 2)

0.12	0.43	0.92	1.14	1.24	1.61	1.93	2.38	4.51	5.09
6.79	7.64	8.45	11.90	11.94	13.01	13.25	14.32	17.47	18.10
18.66	19.23	24.39	25.01	26.41	26.80	27.75	29.69	29.84	31.65
32.64	35.00	40.70	42.34	43.05	43.40	44.36	45.40	48.14	49.10
49.44	51.17	58.62	60.29	72.13	72.22	72.25	72.29	85.20	89.52

by

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} [y(t_i; \boldsymbol{\theta}) - y_i]^2 \,, \tag{3.61}$$

where $y(t_i; \boldsymbol{\theta})$ uses the vector parameter $\boldsymbol{\theta}$ and y_i is the corresponding value obtained from the data. The optimization can be carried out using any standard optimization package. The least-squares method not only furnishes us with parameter estimates, but also helps to select a Weibull model. If one of the potential candidates has a value for $J(\boldsymbol{\theta})$ which is considerably smaller than that for the other models, then this can undoubtedly be accepted as the most appropriate model for modeling the given data set. If two or more Weibull models give rise to roughly the same value of $J(\boldsymbol{\theta})$, one would need to examine additional properties of the WPP plots to decide on the final model. Other approaches such as bootstrap and jackknife may be employed for the final selection. For more on this, see *Murthy* et al. [3.3, 47].

A couple of comments on step 3 of our empirical modeling may be in order. There are many statistical tests for validating a model. These generally require data that is different from the data used for model selection and parameter estimation. A smaller data set may pose a problem, as there will be no separate data left after model selection and parameter estimation. Various solutions have been proposed in the case of a small data set. We refer the readers to the book by *Meeker* and *Escobar* [3.48] for further details.

Example 3.1:
50 items are tested to failure. The failure times are automatically recorded and given in Table 3.1. The Weibull probability plot indicates that the two-parameter Weibull model is not appropriate.

Based on the plot, we observe that the plot shows a decreasing slope at the beginning, and an increasing slope at the end. This is an indication of a distribution with a bathtub-shaped failure-rate function. Some of the models of type II or type III can be used. One possibility is to fit the early part and last part of the plot with separate lines (Fig. 3.4). This will result in a two-fold competing risk model (3.48).

3.6 Applications

3.6.1 Applications in Reliability

Product reliability depends on the design, development and manufacturing decisions made prior to the launch (pre-launch stage) of the product and it in turn affects the failures when the product is put into operation after launch (post-launch stage).

The pre-launch stage involves several phases. In the feasibility phase, study is carried out using the specified target value for product reliability. During the design phase, product reliability is assessed in terms of part and component reliabilities. Product reliability increases as the design is improved. However, this improvement has an upper limit. If the target value is below this limit, then the design using available parts and components achieves the desired target value. If not, then a program to improve the reliability through test–fix–test cycles is carried out during the development phase. Here the prototype is tested until a failure occurs and the causes of the failure are analyzed. Based on this, design changes are made to overcome the identified failure causes. This process is continued until the reliability target is achieved. The reliability of the items produced during manufacturing tends to vary from the design target value due to variations resulting from the manufacturing process. Through proper process and quality control during the manufacturing phase, these variations are controlled.

In the post-launch stage, the reliability of an item decreases due to deterioration resulting from age and/or usage. This deterioration is affected by several factors, including the environment, operating conditions and maintenance. The rate of deterioration can be controlled through effective preventive maintenance actions. Poor reliability results in higher maintenance cost for the

Table 3.2 A sample of reliability applications

Author(s)	Topic
Weibull [3.5]	Yield strength of steel, fatigue life of steel
Keshevan et al. [3.49]	Fracture strength of glass
Sheikh et al. [3.50]	Pitting corrosion in pipes
Quereshi and *Sheikh* [3.51]	Adhesive wear in metals
Durham and *Padget* [3.52]	Failure of carbon-fiber composites
Almeida [3.53]	Failure of coatings
Fok et al. [3.54]	Failure of brittle materials
Newell et al. [3.55]	Failure of composite materials
Li et al. [3.56]	Concrete components

buyer. It also leads to higher warranty cost for the manufacturer resulting from the cost of rectifying all failures within the warranty period subsequent to the sale of the product.

Most products are composed of several components and the failure of the product is due to failure of its components. Weibull models (the two- and three-parameter models as well as a variety of types I–III Models) have been used to model the failures of many components and the literature is vast. A very small sample of the literature follows.

These models allow the determination of product reliability in terms of component reliabilities during the design phase.

During the development phase, it is often necessary to use accelerated testing to hasten the process leading to component failures. A variety of type IV models have been used for the design of experiments to carry out this testing. For more on such models, see *Nelson* [3.39] and *Meeker* and *Escobar* [3.48].

The improvement in reliability (also referred to as reliability growth) during the development phase has been modeled in many different ways. *Duane* [3.57] used a Weibull intensity model formulation to model the improvement in failure rate as a function of the development time. The breakthroughs leading to improvements can be viewed as random points along the time axis. *Crow* [3.58] modeled this by a Weibull power-law process (type VI Weibull model). For models that are modification of this model, see *Murthy* et al. [3.3].

In the manufacturing phase, the fraction of nonconforming items is small when the process is in-control while it increases significantly when the process goes out-of-control. The Weibull distribution has been used to model the in-control duration in the design of control charts to detect the change from in-control to out-of-control. For more on this, see *Rahim* [3.59], *Costa* and *Rahim* [3.60], and *Chen* and *Yang* [3.61]. *Nelson* [3.62] deals with control charts for items with Weibull failure distributions where conforming and nonconforming items differ in the scale parameter but have the same shape parameter. *Murthy* et al. [3.63] and *Djamaludin* et al. [3.64] deal with lot production and look at optimal lot size to control the occurrence of nonconforming items.

When the failure rate has a bathtub shape, it is prone to early failure. For products modeled by Weibull models exhibiting bathtub failure rates, burn-in is a technique that can be used to weed out such failures and improve product reliability before it is released for sale. For more on burn-in, see, e.g., *Kececioglu* and *Sun* [3.65].

Warranty cost analysis for products with a Weibull failure distribution has received a lot of attention. For more on this, see, e.g., *Blischke* and *Murthy* [3.66, 67] and *Murthy* and *Djamaludin* [3.68].

Preventive maintenance of products with a Weibull failure distribution has received considerable attention. In the age policy, an item is replaced preventively when it reaches some specified age and *Tadikmalla* [3.69] deals with this in the context of the Weibull failure distribution. In the block policy, items are replaced preventively at set clock times and *Blischke* and *Murthy* [3.70] deals with this in the context of the Weibull failure distribution.

3.6.2 Applications in Other Areas

Weibull distribution has been used as a model in diverse disciplines to study many different issues. There are

Table 3.3 A sample of other applications

Discipline	Topic	Author(s)
Geophysics	Wind-speed data analysis	*Al-Hasan* [3.71]
	Earthquake magnitude	*Huillet* and *Raynaud* [3.72]
	Volcanic occurrence data	*Bebbington* and *Lai* [3.73]
	Low-flow analysis	*Durrans* [3.74]
	Regional flood frequency	*Heo* et al. [3.75]
Food science	Sterility in thermal preservation method	*Mafart* et al. [3.76]
Social science	Unemployment duration	*Roed* and *Zhang* [3.77]
Environment	Environment radioactivity	*Dahm* et al. [3.78]
Nature	Ecological application	*Fleming* [3.79]
Medical science	Survival data	*Carroll* [3.80]

several thousand papers and we give a very small sample of this vast literature

3.6.3 Weibull Analysis Software

In any analysis of statistical data, computer software is required. In addition to standard statistical software such as MINITAB, SPSS, SAS, etc., or spreadsheet software such as Excel, some specialized Weibull analysis software are also available. Below is a list of some common ones.

- Weibull++ 6 by ReliaSoft Corp., Tucson, AZ; http://weibull.reliasoft.com/
- Relex Weibull by Relex Software Corp., Greensburg, PA; http://www.relexsoftware.com/products/weibull.asp
- WeibullPro by Isograph Inc., Newport Beach, CA; http://www.isograph-software.com/avsoverwbl.htm
- WinSMITH Weibull by Barringer & Associates, Inc., Humble, TX; http://www.barringer1.com/wins.htm

References

3.1 A. J. Jr. Hallinan: A review of the Weibull distribution, J. Qual. Technol. **25**, 85–93 (1993)
3.2 N. L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions*, Vol. 1, 2nd edn. (Wiley, New York 1994)
3.3 D. N. P. Murthy, M. Xie, R. Jiang: *Weibull Models* (Wiley, New York 2003)
3.4 W. Weibull: A statistical theory of the strength of material, Ing. Vetenskapa Acad. Handlingar **151**, 1–45 (1939)
3.5 W. Weibull: A statistical distribution function of wide applicability, J. Appl. Mech. **18**, 293–296 (1951)
3.6 M. Fréchet: Sur la loi de probabilité de l'écart maximum, Ann. Soc. Polonaise Math. Cracovie **6**, 93–116 (1927)
3.7 R. A. Fisher, L. M. C. Tippett: Limiting forms of frequency distribution of the largest or smallest member of a sample, Proc. Cambridge Philos. Soc. **24**, 180–190 (1928)
3.8 N. L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions*, Vol. 2, 2nd edn. (Wiley, New York 1995)
3.9 W. J. Zimmer, J. B. Keats, F. K. Wang: The Burr XII distribution in reliability analysis, J. Qual. Technol. **30**(4), 386–394 (1998)
3.10 C. D. Lai, M. Xie: Relative ageing for two parallel systems and related problems, Math. Comp. Model. **38**, 1339–1345 (2003)
3.11 M. Xie, C. D. Lai: On the increase of the expected lifetime by parallel redundancy, Asia Pacific J. Oper. Res. **13**, 171–179 (1996)
3.12 J. F. Lawless: *Statistical Models and Methods for Lifetime Data* (Wiley, New York 1982)
3.13 W. Nelson: *Applied Life Data Analysis* (Wiley, New York 1982)
3.14 M. Liao, T. Shimokawa: A new goodness-of-fit test for Type-I extreme-value, 2-parameter Weibull distributions with estimated parameters, J. Stat. Comput. Sim. **64**, 23–48 (1999)
3.15 W. Nelson, V. C. Thompson: Weibull probability papers, J. Qual. Technol. **3**, 140–146 (1971)
3.16 W. Nelson: Theory and application of hazard plotting for censored failure data, J. Qual. Technol. **14**, 935–966 (1972)

3.17 J. D. Kalbfleisch, R. L. Prentice: *The Statistical Analysis of Failure Data* (Wiley, New York 1980)
3.18 A. C. Cohen, B. J. Whitten, Y. Ding: Modified moment estimation for the three-parameter Weibull distribution, J. Qual. Technol. **16**, 159–167 (1984)
3.19 S. H. Zanakis, J. Kyparisis: A review of maximum likelihood estimation methods for the three-parameter Weibull distribution, J. Stat. Comp. Simul. **25**, 53–73 (1986)
3.20 B. Dodson: *Weibull Analysis* (ASQC Quality, Milwaukee 1994) see also 3.81
3.21 M. Xie, C. D. Lai, D. N. P. Murthy: Weibull-related distributions for modelling of bathtub shaped failure rate functions. In: *Mathematical and Statistical Methods in Reliability*, ed. by B. H. Lindqvist, K. A. Doksum (World Scientific, Singapore 2003) pp. 283–297
3.22 C. D. Lai, L. Y. Zhang, M. Xie: Mean residual life and other properties of Weibull related bathtub shape failure rate distributions, Int. J. Reliab. Qual. Saf. Eng. **11**, 113–132 (2004)
3.23 D. N. P. Murthy, J. Baik, M. Bulmer, R. J. Wilson: Two-dimensional modelling of failures. In: *Springer Handbook Of Engineering Statistics*, ed. by H. Pham (Springer, Heidelberg, Berlin 2004)
3.24 A. W. Marshall, I. Olkin: A new method for adding aparameter to afamily of distributions with application to the exponential and Weibull families, Biometrika **84**, 641–652 (1997)
3.25 C. D. Lai, M. Xie, M. D. N. P. Murthy: Modified Weibull model, IEEE Trans. Reliab. **52**, 33–37 (2003)
3.26 G. S. Mudholkar, D. K. Srivastava: Exponentiated Weibull family for analyzing bathtub failure-rate data, IEEE Trans. Reliab. **42**, 299–302 (1993)
3.27 R. Jiang, D. N. P. Murthy: The exponentiated Weibull family: A graphical approach, IEEE Trans. Reliab. **48**, 68–72 (1999)
3.28 J. A. Kies: *The Strength of Glass*, Report, Vol. 5093 (Naval Research Lab., Washington 1958)
3.29 M. Xie, T. N. Goh, Y. Tang: A modified Weibull extension with bathtub-shaped failure rate function, Reliab. Eng. Syst. Saf. **76**, 279–285 (2002)
3.30 R. M. Smith, L. J. Bain: An exponential power life-testing distribution, Commun. Stat. **4**, 469–481 (1975)
3.31 Z. Chen: A new two-parameter distribution with bathtub shape or increasing failure rate function, Stat. Probab. Lett. **49**, 155–161 (2000)
3.32 R. Jiang, D. N. P. Murthy: Modeling failure data by mixture of two Weibull distributions: A graphical approach, IEEE Trans. Reliab. **44**, 477–488 (1995)
3.33 R. Jiang, D. N. P. Murthy: Mixture of Weibull distributions-parametric characterization of failure rate function, Stochastic Models Data Anal. **14**, 47–65 (1998)
3.34 J. Gurland, J. Sethuraman: How pooling failure data may reverse increasing failure rates, J. Am. Stat. Ass. **90**, 1416–1423 (1995)
3.35 R. Jiang, D. N. P. Murthy: Parametric study of competing risk model involving two Weibull distributions, Int. J. Reliab. Quality Saf. Eng. **4**, 17–34 (1997)
3.36 R. Jiang, D. N. P. Murthy: Parametric study of multiplicative model involving two Weibull distributions, Reliab. Eng. Sys. Saf. **55**, 217–226 (1997)
3.37 R. Jiang, D. N. P. Murthy: Reliability modeling involving two Weibull distributions, Reliab. Eng. Sys. Saf. **47**, 187–198 (1995)
3.38 R. Jiang, M. Zuo, D. N. P. Murthy: Two sectional models involving two Weibull distributions, Int. J. Reliab. Qual. Saf. Eng. **6**, 103–122 (1999)
3.39 W. Nelson: *Accelerated Testing* (Wiley, New York 1990)
3.40 T. Nakagawa, S. Osaki: The discrete Weibull distribution, IEEE Trans. Reliab. **24**, 300–301 (1975)
3.41 W. E. Stein, R. Dattero: A new discrete Weibull distribution, IEEE Trans. Reliab. **33**, 196–197 (1984)
3.42 W. J. Padgett, J. D. Spurrier: Discrete failure models, IEEE Trans. Reliab. **34**, 253–256 (1985)
3.43 W. M. Bassin: A Bayesian optimal overhaul model for the Weibull restoration process, J. Am. Stat. Ass. **68**, 575–578 (1973)
3.44 L. J. Bain, M. Engelhart: *Statistical Analysis of Reliability and Life-testing Models*, 2nd edn. (Marcel Dekker, New York 1991)
3.45 D. R. Cox, D. D. Oakes: *Analysis of Survival Data* (Chapman Hall, New York 1984)
3.46 N. Yannaros: Weibull renewal processes, Ann. Inst. Stat. Math. **46**, 641–648 (1994)
3.47 D. N. P. Murthy, M. Bulmer, J. E. Eccleston: Weibull modelling, Reliab. Eng. Sys. Saf. **86**, 257–267 (2004)
3.48 W. Q. Meeker, L. A. Escobar: *Statistical Methods for Reliability Data* (Wiley, New York 1998)
3.49 K. Keshevan, G. Sargent, H. Conrad: Statistical analysis of the Hertzian fracture of pyrex glass using the Weibull distribution function, J. Mater. Sci. **15**, 839–844 (1980)
3.50 A. K. Sheikh, J. K. Boah, D. A. Hansen: Statistical modelling of pitting corrosion and pipeline reliability, Corrosion **46**, 190–196 (1990)
3.51 F. S. Queeshi, A. K. Sheikh: Probabilistic characterization of adhesive wear in metals, IEEE Trans. Reliab. **46**, 38–44 (1997)
3.52 S. D. Durham, W. J. Padgett: Cumulative damage model for system failure with application to carbon fibers and composites, Technometrics **39**, 34–44 (1997)
3.53 J. B. Almeida: Application of Weilbull statistics to the failure of coatings, J. Mater. Proces. Technol. **93**, 257–263 (1999)
3.54 S. L. Fok, B. C. Mitchell, J. Smart, B. J. Marsden: A numerical study on the application of the Weibull theory to brittle materials, Eng. Fract. Mech. **68**, 1171–1179 (2001)
3.55 J. A. Newell, T. Kurzeja, M. Spence, M. Lynch: Analysis of recoil compressive failure in high per-

formance polymers using two-, four-parameter Weibull models, High Perform. Polym. **14**, 425–434 (2002)
3.56 Q. S. Li, J. Q. Fang, D. K. Liu, J. Tang: Failure probability prediction of concrete components, Cement Concrete Res. **33**, 1631–1636 (2003)
3.57 J. T. Duane: Learning curve approach to reliability monitoring, IEEE Trans. Aerosp. **40**, 563–566 (1964)
3.58 L. H. Crow: Reliability analysis for complex systems. In: *Reliability and Biometry*, ed. by F. Proschan, R. J. Serfling (SIAM, Philadelphia 1974) pp. 379–410
3.59 M. A. Rahim: Economic design of $\overline{X}$ charts assuming Weibull in-control times, J. Qual. Technol. **25**, 296–305 (1993)
3.60 A. F. B. Costa, M. A. Rahim: Economic design of X and R charts under Weibull shock models, Qual. Reliab. Eng. Int. **16**, 143–156 (2000)
3.61 Y. S. Chen, Y. M. Yang: Economic design of x-control charts with Weibull in-control times, when there are multiple assignable causes, Int. J. Prod. Econ. **77**, 17–23 (2002)
3.62 P. R. Nelson: Control chart for Weibull process, IEEE Trans. Reliab. **28**, 283–288 (1979)
3.63 D. N. P. Murthy, I. Djamaludin, R. J. Wilson: Product warranty and quality control, Qual. Reliab. Eng. **9**, 431–443 (1993)
3.64 I. Djamaludin, D. N. P. Murthy, R. J. Wilson: Lot-sizing and testing for items with uncertain quality, Math. Comp. Model. **22**, 35–44 (1995)
3.65 D. Kececioglu, F. B. Sun: *Burn-in Testing: Its Quantification and Optimization* (Prentice Hall, Upper Saddle River 1997)
3.66 W. R. Blischke, D. N. P. Murthy: *Product Warranty Handbook* (Marcel Dekker, New York 1994)
3.67 W. R. Blischke, D. N. P. Murthy: *Warranty Cost Analysis* (Marcel Dekker, New York 1996)
3.68 D. N. P. Murthy, I. Djamaludin: Product warranty: A review, Int. J. Prod. Econ. **79**, 231–260 (2002)
3.69 P. R. Tadikmalla: Age replacement policies for Weibull failure rates, IEEE Trans. Reliab. **29**, 88–90 (1980)
3.70 W. R. Blischke, D. N. P. Murthy: *Reliability* (Wiley, New York 2000)
3.71 M. Al-Hasan, R. R. Nigmatullin: Identification of the generalized Weibull distribution in wind speed data by the Eigen-coordinates method, Renewable Energ. **28**(1), 93–110 (2003)
3.72 T. Huillet, H. F. Raynaud: Rare events in a log-Weibull scenario—Application to earthquake magnitude data, Eur. Phys. J. B **12**, 457–469 (1999)
3.73 M. S. Bebbington, C. D. Lai: On nonhomogeneous models for volcanic eruptions, Math. Geol. **28**, 585–599 (1996)
3.74 S. R. Durrans: Low-flow analysis with aconditional Weibull tail model, Water Resour. Res. **32**, 1749–1760 (1996)
3.75 J. H. Heo, D. C. Boes, J. D. Salas: Regional flood frequency analysis based on a Weibull model: Part 1. Estimation, asymptotic variances, J. Hydrol. **242**, 157–170 (2001)
3.76 P. Mafart, O. Couvert, S. Gaillard, I. Leguerinel: On calculating sterility in thermal preservation methods: application of the Weibull frequency distribution model, Int. J. Food Microbiol. **72**, 107–113 (2002)
3.77 K. Roed, T. Zhang: A note on the Weibull distribution and time aggregation bias, Appl. Econ. Lett. **9**, 469–472 (2002)
3.78 H. Dahm, J. Niemeyer, D. Schroder: Application of the Weibull distribution to describe the vertical distribution of cesium-137 on a slope under permanent pasture in Luxembourg, J. Environ. Radioac. **63**, 207–219 (2002)
3.79 R. A. Fleming: The Weibull model and an ecological application: describing the dynamics of foliage biomass on Scots pine, Ecol. Model. **138**(1-3), 309–319 (2001)
3.80 K. J. Carroll: On the use and utility of the Weibull model in the analysis of survival data, Control. Clin. Trials **24**, 682–701 (2003)
3.81 N. L. Johnson, S. Kotz, N. Balakrishnan: *Continuos Univariate Distributions*, Vol. 1, 2nd edn. (Wiley, New York 1994) pp. 657–661

4. Characterizations of Probability Distributions

A characterization is a certain distributional or statistical property of a statistic or statistics that uniquely determines the associated stochastic model. This chapter provides a brief survey of the huge literature on this topic. Characterizations based on random (complete or censored) samples from common univariate discrete and continuous distributions, and some multivariate continuous distributions are presented. Characterizations that use the properties of sample moments, order statistics, record statistics, and reliability properties are reviewed. Applications to simulation, stochastic modeling and goodness-of-fit tests are discussed. An introduction to further resources is given.

Suppose X is the population random variable (RV) with cumulative distribution function (CDF) $F(x) = \Pr(X \le x)$ from which the data are generated according to a specified sampling scheme. Let $\mathcal{F}$ be the family of probability distributions used as an initial model to describe X and $\mathcal{F}_0$ be a subclass of $\mathcal{F}$ that is of interest to the modeler. If $T = T(X)$ is a statistic arising from F, and if a certain distributional property it possesses implies that $F \in \mathcal{F}_0$, then this property of $T(X)$ produces a characterization of $\mathcal{F}_0$. If the concerned property of $T(X)$ holds if and only if $F \in \mathcal{F}_0$, then we have a complete characterization of $\mathcal{F}_0$ and it is particularly helpful when this subfamily has a single member or members from a single parametric family of distributions. This is the essence of the abundant literature on this fascinating area of characterizations, produced by probabilists and mathematical statisticians, mostly over the past half-century.

In Sect. 4.1 we describe characterizing functions and their role in characterizations. Various types of data settings and associated characterizing properties discussed in the literature are reviewed in Sect. 4.2. A general classification of characterization results is given in Sect. 4.3. Treatment of continuous distributions begins with that of the exponential in Sect. 4.4, and continues with the normal in Sect. 4.5, and other distributions in Sect. 4.6. Characterizations for the Poisson distribution and Poisson process are given in Sect. 4.7, and Sect. 4.8 contains characterizations for other common discrete distributions. A brief look at the characterizations of multivariate distributions is provided in Sect. 4.9, where some characterizations based on conditional specification are presented. Special attention is drawn to the Marshall–Olkin bivariate exponential model and the multivariate normal distribution. Stability of characterization results is discussed in Sect. 4.10 and applications of characterizations in Sect. 4.11. The last section (Sect. 4.12) contains a listing of major resources.

4.1 Characterizing Functions

There are several functions associated with a probability distribution that uniquely identify it. We call these characterizing functions and describe a few of them.

4.1.1 Cumulative Distribution Function (CDF)

The CDF $F(x)$ of a RV X, defined for all real x, describes $P(X \in A)$ for any Borel set A on the real line. It is right-continuous, nondecreasing, and $F(-\infty) = 0$ and $F(+\infty) = 1$. If X is a discrete RV, $\Pr(X = x) = F(x) - F(x-) > 0$ for any possible value x. A closely associated function, the *survival function* (SF), is defined by

$$S(x) = \Pr(X \geq x) = 1 - F(x-)\,. \tag{4.1}$$

4.1.2 Probability Density Function (PDF)

The PDF $f(x)$ is a nonnegative function with the property that, for any Borel set A, $\Pr(X \in A)$ can be obtained by either summing or integrating $f(x)$ over A. When X is absolutely continuous, $f(x) = F'(x)$ almost everywhere (a. e.). In many cases, the PDF provides the most convenient way to describe the probability assignment.

4.1.3 Quantile Function

Also known as the inverse CDF, it is often defined as

$$F^{-1}(u) = \inf\{x : F(x) \geq u\},\ 0 < u < 1\,. \tag{4.2}$$

The quantile function is nondecreasing and, in this form, is left-continuous. It describes the probability assignment in terms of the quantiles of the distribution. When X is absolutely continuous, $F^{-1}(u)$ is differentiable with derivative $1/f[F^{-1}(u)]$.

The *probability integral transformation* $U = F(X)$ transforms any continuous RV X with CDF $F(x)$ into a standard uniform RV U that has PDF $f(u) = 1$, $0 < u < 1$. For *any* RV X with CDF $F(x)$,

$$X \stackrel{d}{=} F^{-1}(U)\,, \tag{4.3}$$

where $\stackrel{d}{=}$ stands for equality in distribution. This valuable distributional identity is helpful in simulating arbitrary RVs and also plays an important role in the theoretical developments associated with moments of X and of order statistics generated from random samples.

For a nonnegative RV with finite mean $\mu = \mathrm{E}(X) = \int_0^1 F^{-1}(u)\,\mathrm{d}u$, the quantile function yields the *Lorenz curve* used for describing the discrepancy in income distributions. The Lorenz curve is described by the function

$$L(t) = \frac{1}{\mu}\int_0^t F^{-1}(u)\,\mathrm{d}u,\quad 0 \leq t \leq 1\,. \tag{4.4}$$

It identifies F^{-1} and consequently F, up to the scale parameter.

4.1.4 Characteristic Function (CF) and Other Generating Functions

For any RV X, a CF exists and is defined by $\psi(t) = \mathrm{E}[\exp(\mathrm{i}tX)]$, where t is real and $\mathrm{i} = \sqrt{-1}$. This complex-valued function (it is real for F symmetric about 0) uniquely determines the CDF ([4.1], p. 104). It has played a major role in the proof of the central limit theorem and in numerous characterizations, especially of the normal distribution. An associated function, the moment generating function (MGF), given by $\mathrm{E}[\exp(tX)]$, may not exist for $t \neq 0$. If the MGF exists for t in a neighborhood of 0, then moments of X of all orders exist. Furthermore, the jth moment of X, $\mathrm{E}(X^j)$, can be obtained by a Taylor-series expansion of either

the CF or the MGF of X. A one-to-one function of the MGF is its logarithm, the cumulant generating function. A discrete distribution with support over nonnegative integers is uniquely determined by its probability generating function, given by $\mathrm{E}\{s^X\}$, defined at least for $|s| \le 1$ ([4.1], p. 101).

4.1.5 Reliability Considerations

Several characterizing functions that are based on reliability properties have appeared in the area of life-testing experiments. There are many classes of distributions in the reliability literature that are based on the properties of such functions [4.2]. We introduce three such functions where we assume X to be nonnegative for practical reasons, and to be absolutely continuous for convenience.

Failure Rate and Hazard Functions

The failure (or hazard) rate function is defined for all x in the support of the RV X and is given by

$$h(x) = \frac{f(x)}{S(x)} = \frac{f(x)}{1 - F(x)} , \tag{4.5}$$

where $S(x)$ is the SF defined in (4.1), and in this case, equals $1 - F(x)$. Clearly,

$$H(x) \equiv \int_0^x h(w)\,\mathrm{d}w = -\log[1 - F(x)] , \tag{4.6}$$

and thus $h(x)$ uniquely determines $F(x)$. Further, $H(x)$, the integrated failure-rate function, also called the *hazard function*, identifies $F(x)$ as well, through the relationship $F(x) = 1 - \exp[-H(x)]$. The family of distributions for which the failure rate (FR) $h(x)$ is increasing (decreasing) is the IFR (DFR) family. If $H(x)/x$ is increasing (decreasing) one obtains the IFRA (DFRA) family since the average failure rate in that case will be increasing (decreasing).

Mean Residual Life (MRL) Function

The MRL is of practical interest in life-testing experiments and is defined whenever $\mathrm{E}(X)$ is finite. For a nonnegative X, with $t > 0$, it is given by

$$\begin{aligned} m(t) &\equiv \mathrm{E}(X - t | X > t) \\ &= \frac{1}{1 - F(t)} \int_t^\infty [1 - F(w)]\,\mathrm{d}w , \end{aligned} \tag{4.7}$$

if the CDF F is continuous. Equation (4.7) can be used to recover $F(x)$ from $m(t)$. In fact, when F is continuous, and $F^{-1}(0) = 0$,

$$1 - F(x) = \frac{m(0)}{m(x)} \exp\left(- \int_0^x \frac{\mathrm{d}y}{m(y)} \right) , x > 0 , \tag{4.8}$$

where $m(0) = \mathrm{E}(X)$. Another closely related characterizing function is the truncated mean $\mathrm{E}(X | X > t) \equiv t + m(t)$, sometimes referred to as the conditional tail expectation. Increasing (decreasing) $m(t)$ produces the family of IMRL (DMRL) life distributions. Yet another characterizing function is the total time on test transform ([4.3], p. 91), defined as

$$\tau(t) = \int_0^{F^{-1}(t)} [1 - F(x)]\,\mathrm{d}x, 0 \le t \le 1 . \tag{4.9}$$

This is concave if and only if (iff) F is IFR.

Characterization of a probability distribution then refers to the identification of any of $F(x)$, $F^{-1}(x)$, $f(x)$, $\psi(t)$, $h(x)$, $H(x)$, $m(t)$, $\tau(t)$, or of families for which these functions possess a certain property.

Example. Consider the exponential distribution with rate parameter λ whose PDF is given by

$$f(x) = \lambda\,\mathrm{e}^{-\lambda x}, x \ge 0 . \tag{4.10}$$

If (4.10) holds, we say that X is an $\exp(\lambda)$ RV. For such an RV, the CDF is given by $F(x) = 1 - \exp(-\lambda x)$, $x \ge 0$; the quantile function is given by $F^{-1}(u) = -\log(1-u)/\lambda$, $0 \le u \le 1$; the characteristic function by $\psi(t) = 1/(1 - \mathrm{i}t/\lambda)$; the failure-rate function by $h(x) = \lambda$, $x \ge 0$; the hazard function by $H(x) = \lambda x$, $x \ge 0$; the MRL function by $m(t) = 1/\lambda$, $t \ge 0$; and the total time on test transform by $\tau(t) = t/\lambda$, $t \ge 0$.

4.2 Data Types and Characterizing Conditions

A characterization result is based on the assumed model for the data and the specified property of a particular statistic of interest. We now describe some common scenarios.

4.2.1 Data Models

The most basic data set consists of a single observation X from the CDF $F(x)$. Next, one may encounter either

a random sample $X_1, \ldots, X_n$ from $F(x)$, i. e., the X_j are independent, identically distributed (IID) RVs with CDF $F(x)$. Or the X_j are just independent and their CDFs belong to a family $\mathcal{F}_0$. Type II right-censored samples from (absolutely) continuous distributions that appear naturally in life-testing experiments are considered as well.

Let $X_{1:n} \le \cdots \le X_{k:n} \le \cdots \le X_{n:n}$ be the order statistics of the random sample [4.3]. Then the CDF of $X_{k:n}$ is given by

$$\sum_{j=k}^{n} \binom{n}{j} F^j(x)[1-F(x)]^{n-j}$$

$$= \frac{n!}{(k-1)!(n-k)!} \int_0^{F(x)} t^{k-1}(1-t)^{n-k}\,\mathrm{d}t\,, \qquad (4.11)$$

and the joint PDF of $X_{1:n}, \cdots, X_{k:n}$, a type II right censored sample, is given by

$$\frac{n!}{(n-k)!} f(x_1)\cdots f(x_k)[1-F(x_k)]^{n-k},$$

$$x_1 < \cdots < x_k\,. \qquad (4.12)$$

Models that use the properties of upper or lower record values for characterizations do exist. For an infinite sequence $\{X_j, j \ge 1\}$ from a continuous parent, for $j \ge 1$, X_j is called an *upper record value* of this sequence if $X_j = \max(X_1, \ldots, X_j)$ [4.4]. By convention, X_1 is the first upper record value (or reference value). The joint PDF of the first m upper record values, $R_1, \ldots, R_m$, is given by

$$f(r_1, \ldots, r_m) = f(r_m) \prod_{j=1}^{m-1} h(r_j) \quad r_1 < \cdots < r_m\,, \qquad (4.13)$$

where $h(x)$ is given by (4.5). From (4.12) and (4.13), we can, respectively, conclude that (see, e.g., [4.4], p. 114), for a continuous parent

$$\Pr(X_{k+1:n} > y | X_{k:n} = x)$$

$$= [\Pr(X > y | X > x)]^{n-k}, x < y\,, \qquad (4.14)$$

and

$$\Pr(R_{m+1} > y | R_m = x) = \Pr(X > y | X > x)$$

$$= \frac{1-F(y)}{1-F(x)}, x < y\,. \qquad (4.15)$$

4.2.2 Characterizing Conditions

There are many types of characterizing properties. For example, the property could be based on identical distribution of two statistics, or independence of them, or on constancy of regression of one on the other. Alternatively, the characterization may be based on properties of admissibility and optimality of certain estimators based on random samples [4.5]. Characterizations based on fixed or random sample sizes and on damaged/missing observations do exist. Basically, the assumed condition is shown to yield a functional equation satisfied by a characterizing function, leading to the identification of the parent distribution. Other conditions involve recurrence relations satisfied by the ordinary moments, or moments of order statistics. There are characterizations based on certain inequalities for the moments, Fisher information or entropy measures, where equality holds for a unique family of distributions.

4.2.3 General Techniques

For the normal distribution, identification via the characteristic function is the common approach. The proofs involve results on complex variables. Numerous results for the exponential or geometric distributions have been based on the SF and rely on the Cauchy functional equation (CFE) ([4.6], Sect. 1.1), namely,

$$g_0(x+y) = g_0(x)g_0(y), \text{ for all } x, y \ge 0\,. \qquad (4.16)$$

The only continuous solution is the exponential SF. The well-known lack-of-memory property (LMP) of the exponential distribution, namely

$$\Pr(X > x+y | X > x) = \Pr(X > y),$$

$$\text{for all } x, y \ge 0\,, \qquad (4.17)$$

leads to (4.16). In (4.17), the exponential SF $\Pr(X > x) = \exp(-\lambda x)$ is recovered even if the equation holds for all $x \ge 0$ and for two y values y_1, y_2 such that y_1/y_2 is irrational.

In recent years, the integrated CFE (ICFE) whose solution goes back to Deny's Theorem ([4.6, 7], Chapt. 2) has been used for several characterizations of the exponential and geometric distributions. We say that g_1 satisfies an ICFE if it is a function defined on $[0, \infty)$ and satisfies the condition (4.18) (see below). The following characterization result due to *Lau* and *Rao* [4.8] is taken from [4.7] (p. 29).

Theorem 4.1

Let g be a nonnegative locally integrable function on $[0, \infty)$ that is not a function identically equal to 0 a.e. ν_0, the Lebesgue measure defined on $[0, \infty)$. Suppose g satisfies (4.18) given below where ν is a σ-finite measure

satisfying the condition $\nu(\{0\}) < 1$:

$$g(x) = \int_0^\infty g(x+y)\, \mathrm{d}\nu(y), \qquad \text{a. e. } \nu_0 \text{ for } x \geq 0\,. \tag{4.18}$$

Then either

$$g(x+n\delta) = g(x)b^n, n \geq 0, \qquad \text{a. e. } \nu_0 \text{ for } x \geq 0\,, \tag{4.19}$$

where b is such that $\sum_{n=0}^\infty b^n \nu(\{n\delta\}) = 1$ for some $\delta > 0$, or

$$g(x) \propto \mathrm{e}^{\alpha x} \quad \text{a. e. } \nu_0 \text{ for } x \geq 0\,, \tag{4.20}$$

where α satisfies the condition $\int_0^\infty \mathrm{e}^{\alpha y}\, \mathrm{d}\nu(y) = 1$.

If g_0 satisfying (4.16) is integrable with respect to a σ-finite measure ν_1 on $[0, \infty)$, the integral being a positive quantity c, then it also satisfies (4.18) with $\nu = c^{-1}\nu_1$.

The above result has provided elegant unified proofs of several characterizations of exponential and geometric distributions. See the review in [4.9] and the monograph [4.7]. A nice account of the role of functional equations in probability theory and in characterization theorems is given in [4.6].

A handy technique is based on complete families of functions. Suppose $\{f_n(x),\, n \geq 1\}$ is a sequence such that if, for any integrable function $g(x)$, the condition

$$\int_A g(x) f_n(x)\, \mathrm{d}x = 0 \qquad \text{for all } n \geq 1 \tag{4.21}$$

implies that $g(x) = 0$ a.e. on A, where A is an interval. Then we say that the sequence of functions $f_n(x)$ is complete on A. For example, $f_n(x) = x^n$ is complete on $[0, 1]$ and leads to the characterization of F based on the sequence $\mathrm{E}(X_{n:n})$. A classical completeness result due to Müntz and Szász shows that even an appropriately chosen subsequence would do. (See [4.10] for a good summary.) The *method of intensively monotone operators* is another general method of showing uniqueness of solutions to the functional equations generated by the characterizing conditions [4.11].

An approach involving inequalities such as the Cauchy–Schwarz (or Cramér–Rao) has produced some characterizations. The distribution being characterized corresponds to the case where equality holds. For example,

$$\mathrm{Cor}(X_{j:n}, X_{k:n}) \leq \sqrt{\frac{j(n-k+1)}{k(n-j+1)}},\ j < k\,, \tag{4.22}$$

where "Cor" represents the correlation coefficient, and equality holds iff F is a uniform CDF.

The *method of limit laws*, elaborated in [4.12], is found to be helpful in establishing some characterizations.

4.3 A Classification of Characterizations

We attempt below a general classification of characterization results.

4.3.1 Uniqueness Conditions

These are properties that provide a one-to-one correspondence with the parent CDF F. For example, from (4.11) it is clear that the CDF of $X_{k:n}$ for any fixed k and n identifies $F(x)$, and thus provides a trivial characterization.

A more interesting question is considered in the *classical moment problem* [4.13]. It is concerned with the determination of the CDF $F(x)$ from the sequence of population moments $\{\mathrm{E}(X^j),\, j \geq 1\}$, which are assumed to exist. Two distinct distributions can have the same moment sequence. Under certain conditions, however, the associated CDF is unique. One such condition is that

$$\sum_{j=1}^\infty \frac{\mathrm{E}(X^j)}{j!} t^j \tag{4.23}$$

is absolutely convergent for some $t > 0$ ([4.1], p. 106).

In the context of order statistics, the moment problem has the goal of identifying F based on the moment sequence $\{\mathrm{E}(X_{k(n):n}),\, 1 \leq k(n) \leq n,\, n \geq 1\}$. There is no loss of generality in choosing $k(n)$ [or $n - k(n)$] to be a constant. It is known that the subsequence $\{\mathrm{E}(X_{n_j:n_j}),\, n_1 < n_2 < \cdots,\, j \geq 1\}$ characterizes F provided that $\sum_{j=1}^\infty 1/n_j$ diverges [4.14]. Such a subsequence will not suffice when we consider $\mathrm{E}(R_n)$—the moment sequence of record values. One needs the entire sequence and identification can be

achieved only within the family of continuous distributions ([4.4], Sect. 4.2.1).

It is clear from (4.15) that $E(R_{n+1} - R_n | R_n = x) = m(x)$ a.s., and hence, in view of (4.8), one can recover $F(x)$ from the regression function $E(R_{n+1}|R_n)$. Similarly $E(X_{k+1:n}|X_{k:n})$ or even $E[h(X_{k+1:n})|X_{k:n}]$ where $h(.)$ is a real, continuous and strictly monotonic function, would identify F in the family of *arbitrary* distributions [4.15].

4.3.2 Characterizations of Families of Distributions

There are characterizations of distributions defined by their reliability properties, of the exponential-family distributions, or of spherically symmetric (multivariate) distributions, and other large families. Such results identify the general properties possessed by the members without actually providing an explicit form for the CDF or other characterizing functions. For example, certain properties of order statistics characterize various classes of life distributions [4.16]. A specific characterization of this sort is the following. Among nonnegative distributions with finite mean, a CDF F is IFR iff $E[(n-k+1)(X_{k:n} - X_{k-1:n})]$ is decreasing in k $(2 \le k \le n)$ for infinitely many n. A one-parameter exponential family is characterized (under some conditions) by the fact that, for a single random sample of size $n \ge 3$, the sample mean $\bar{X}$ is the maximum-likelihood estimator (MLE) of the mean of the distribution (see also [4.17]). Such results are interesting theoretically and provide some reality checks for the model assumptions.

4.3.3 Characterizations of Specific Parametric Families

Of maximal interest in terms of applications and creation of goodness-of-fit tests are the characterizations that identify specific parametric family of distributions. In subsequent sections we will discuss a few such results for some common distributions; in fact, we consider only a tiny subset of the voluminous, ever-growing literature. We only list the characterization results with no details on the other needed conditions, some of which could be very technical and hard to verify. We refrain from proving any of our claims.

4.4 Exponential Distribution

Arnold and *Huang* [4.18], in their survey of characterizations of the exponential distribution, mentioned about 275 citations 10 years ago, and one could safely add 50 more to the list, making it perhaps the most popular distribution on this topic. In recent years results based on the properties of order statistics and record values have been in vogue. In most cases, the condition imposed will result in the CFE or the ICFE discussed in Sect. 4.2. Apart from the exponential characterizations stemming from the general results [e.g., if $E(X_{1:n}) = \lambda/n$, $n \ge 1$, then F must be an exp(λ) CDF], the conditions imposed could be based on: (i) truncated moments or regression functions associated with order statistics or (upper) records, (ii) distributional identities among order statistics, their spacings, or the spacings of records, (iii) independence of certain linear functions of order statistics or records, (iv) reliability properties of order statistics or records, or (v) *geometric compounding*, where the RV of interest is a random sum of IID RVs where the number in the sum is determined by an independent geometric RV.

Consider the order statistics case. For the exp(λ) parent: (a) $X_{j:n}$ and $X_{k:n} - X_{j:n}$, $j < k$, are independent, (b) $nX_{1:n} \stackrel{d}{=} X_1$, $n \ge 2$, and the normalized spacings, $Y_j = (n-j+1)(X_{j:n} - X_{j-1:n})$, $j = 1, \ldots, n$ (with $X_{0:n} = 0$), have the following properties: (c) the Y_j are independent, (d) they are identically distributed as X_1 for all $j = 1, \ldots, n$, and (e) $E(Y_j) = 1/\lambda$, $1 \le j \le n$. Under some mild conditions each of these provides a characterization of the exponential CDF. In (b), it is known that, if the distributional equality holds for two values of n, say n_1 and n_2 such that $\log n_1 / \log n_2$ is irrational, then the exponential RV is identified in the family of nonnegative RVs.

If F is an exponential CDF, from the joint PDF of the record values given in (4.13), it can be shown that, for $j \le m < n$:

1. R_m and $R_n - R_m$ are independent,
2. R_j and $R_n - R_m$ are independent,
3. $E(R_{n+1} - R_n | R_n)$ does not depend on R_n,
4. $\mathrm{Var}(R_{n+1} - R_n | R_n)$ does not depend on R_n,
5. $R_n - R_m$ and R_{n-m} are identically distributed,
6. $E((R_n - R_m)^s | R_j)$ does not depend on R_j,
7. $E(R_n) = n/\lambda$, and
8. $E(R_{n+1}|R_n)$ and $E(R_n|R_{n+1})$ are both linear in conditioning variables.

Each of these provides a characterization of the exponential distribution (possibly with a location shift) in

an appropriately chosen $\mathcal{F}$ (such as continuous CDF). There are many more that dwell on the properties of the spacings of upper record values or order statistics.

Using (4.13) and (4.14), characterizations involving record or order statistic spacings can be linked to those based on truncated distributions. For example, assuming that $\mathrm{E}(R_{n+1} - R_n | R_n)$ does not depend on R_n is basically the same as saying $\mathrm{E}(X - x | X > x)$ is free of x. This means that the MRL is a constant—a characterizing property of the exponential distribution. Conditions such as $X_{j+1:n} - X_{j:n} \stackrel{d}{=} X_{1:n-j}$ or $R_n - R_m \stackrel{d}{=} R_{n-m}$ result in the ICFE (see Theorem 4.1), yielding an exponential characterization in the family of continuous CDFs.

In the context of life-testing experiments, the statistic representing the *total time on test* by time $X_{i:n}$, $T_{i,n} = \sum_{j=1}^{i} Y_j = \sum_{j=1}^{i} X_{j:n} + (n-i)X_{i:n}$, is of considerable interest. For an exponential parent, for $2 \le k \le n$,

$$\left(\frac{T_{1,n}}{T_{k,n}}, \cdots, \frac{T_{k-1,n}}{T_{k,n}} \right) \tag{4.24}$$

behaves like the vector of order statistics from a random sample of size $k-1$ from the standard uniform distribution. The converse is shown to be true assuming that the order-statistic property holds for some k and n such that $5 \le k \le n$ [4.9].

There is a characterization of the exponential distribution based on records that is similar to that of the uniform distribution based on the order statistics [see (4.22)]. If $\mathrm{Var}(R_m)$ and $\mathrm{Var}(R_n)$ are both finite, $\mathrm{Cor}(R_m, R_n)$ does not exceed $\sqrt{m/n}$, $m < n$. Furthermore, the upper bound is attained if F is an exponential CDF, possibly with a location shift.

Numerous characterizations of the exponential distribution exist when $\mathcal{F}$ is restricted to families defined by the reliability properties (such as the *new better/worse than used* families). Characterizations of the exponential distribution arising from queueing models are rare; for a few results, see [4.19].

The survey of characterizations using order statistics by *Gather* et al. [4.20] contains an excellent compendium of results for the exponential distribution based on their properties. Other important references include the monograph by *Azlarov* and *Volodin* specializing on exponential characterizations [4.21], the survey [4.18] mentioned earlier, and [4.22], Chapt. 19.

Remarks. The memoryless property (LMP) of the exponential distribution is also shared by the geometric distribution, which can be extracted as $[X]$ where $[\cdot]$ represents the greatest integer function. This results in many parallel characterizations for the geometric parent among distributions on nonnegative integers (see Sect. 4.8; [4.23]). Homogeneous Poisson processes, characterized by the fact that the inter-arrival times of the events are IID exponential, can be identified by the characterizing properties of the exponential distributions. Also, for any RV X with continuous CDF F, $-\log[1 - F(X)]$ is standard exponential and thus exponential characterizations naturally lead to characterizations of such distributions. For example, results based on exponential spacings will lead to the characterizations of the uniform, power-function or Pareto distributions on the basis of the properties of the ratios of order statistics.

4.5 Normal Distribution

The earliest characterization results were for the normal distribution. In 1923 Pólya showed that, if X_1 and X_2 are IID with finite variance, and $X_1 \stackrel{d}{=} a_1 X_1 + a_2 X_2$, then the X_j must necessarily be normal. *Cramér* showed in 1937 that, if X_1 and X_2 are independent and the sum is assumed to be normal, then each of them must be normal (see [4.24], p. 53). *Skitovich* and *Darmois* established in 1953 the following result:

Theorem 4.2

If $X_1, \ldots, X_n$ are independent and the linear functions

$$L_1 = \sum_{j=1}^{n} a_j X_j, \; L_2 = \sum_{j=1}^{n} b_j X_j \tag{4.25}$$

are independent, then the RVs X_j for which $a_j b_j \neq 0$ must all be normal.

The work on normal characterizations accelerated from the 1950s and it was an extremely active area of research until the late 1970s. Extensive accounts are available, particularly in the influential book on characterizations by *Kagan* et al. [4.5], and also in the monograph on the applications of characteristic functions by *Lukacs* and *Laha* [4.25], all major contributors to the area. A short book devoted to normal characterizations by *Mathai* and *Pederzolli* [4.26] also provides a good account. An excellent brief summary of the normal characterizations and an extensive reference list is provided in [4.22],

Chapt. 13. Another recent account of normal characterizations is provided by the monograph [4.27]. We now record a listing of assorted types of characterizations, mostly those that are easy to describe. These hold under usually mild conditions. Many normal characterizations hold under technical conditions that are hard to verify and often it is difficult to assess their practical implications.

Characterizations

We begin with characterizations based on the properties of a single observation from the parent distribution.

Among absolutely continuous distributions with support $(-\infty, \infty)$ and specified mean and variance, the entropy $-\int_{-\infty}^{\infty} f(x)\log f(x)\mathrm{d}x$ is the largest for the normal distribution ([4.5], p. 410).

Assume $\mathcal{F}$ is the location family of absolutely continuous distributions with support $(-\infty, \infty)$ and differentiable density $f(x-\theta)$, where θ is the location parameter. Then the Fisher information in this location family is given by

$$I_f(\theta) = \int\limits_{f(x)>0} \left(\frac{f'(x)}{f(x)}\right)^2 f(x)\mathrm{d}x\,. \tag{4.26}$$

The distribution in this family with the smallest $I_f(\theta)$ is normal ([4.5], p. 406).

The following properties based on a random sample $X_1, \ldots, X_n$ of size n, characterize the normal parent:

1. X_1 has zero mean and unit variance, and for some $a_1, a_2 \neq 0$, $(a_1X_1 + a_2X_2)^2/(a_1^2 + a_2^2) \stackrel{d}{=} (a_1X_1 - a_2X_2)^2/(a_1^2 + a_2^2) \stackrel{d}{=} \chi^2_{(1)}$ ([4.22], p. 105).
2. X_1 has zero mean and is nondegenerate and with $ab = -1$ and $n = 2$, $\mathrm{E}(X_1 + aX_2|X_1 + bX_2) = \mathrm{E}(X_1 + bX_2|X_1 + aX_2) = 0$ ([4.5], p. 158).
3. The sample mean $\bar{X}$ and sample variance S^2 are independent for some $n \geq 2$ ([4.5], p. 103).
4. For some $n \geq 5$, the vector
$$\left(\frac{X_1 - \bar{X}}{(n-1)S^2}, \cdots, \frac{X_n - \bar{X}}{(n-1)S^2}\right) \tag{4.27}$$
is uniformly distributed on the $(n-2)$-dimensional sphere $\left\{(w_1, \ldots, w_n) : \sum_{i=1}^n w_i = 0, \sum_{i=1}^n w_i^2 = 1\right\}$ ([4.7], p. 142).
5. $\mathcal{F}$ is the location family of distributions $F_0(x-\theta)$, with mean θ, the location parameter, and finite variance, and, for some $n \geq 3$, $\bar{X}$ is admissible under squared error loss among all unbiased estimators of θ ([4.5], p. 228).
6. In the above location family (with finite variance and θ as the mean), take the null hypothesis $H_0 : \theta = 0$ and the alternative $H_1 : \theta > 0$. Suppose that, for some $n \geq 3$, the critical region $\{\bar{X} > c_\alpha\}$ is the uniformly most powerful among all tests with level of significance $\leq \alpha$ for all $\alpha \in (0, 1)$, where $c_\alpha = \max\{c : \Pr(\bar{X} > c|\theta = 0) = \alpha\}$ ([4.5], p. 451).
7. In the above location family with θ as the mean, $\bar{X}_n$ is the MLE of θ for $n = 2, 3$ ([4.5], p. 411).
8. $\mathcal{F}$ is the location-scale family of continuous distributions $F_0[(x-\theta)/\sigma]$, θ real, $\sigma > 0$, and $(\bar{X}, S)$ is a sufficient statistic for (θ, σ) ([4.22], p. 106).
9. In the above location-scale family with mean θ, $\bar{X}_n$ is a best linear unbiased estimation (BLUE) of θ for *all* $n \geq 1$ [4.28].
10. In the linear regression model under the Bayesian framework, the distributions with posterior expectation is linear in data values [4.29].

When the X_j are independent RVs, some of the above results have appropriate generalizations. For example,

1. In (4.25), if the sequences $\{a_j/b_j\}$ and $\{b_j/a_j\}$ for which $a_jb_j \neq 0$ are bounded and as $n \to \infty$ L_1 and L_2 converge with probability 1 to independent RVs, then the X_j for which $a_jb_j \neq 0$ are normal ([4.5], p. 94).
2. If $X_1 \stackrel{d}{=} L_1$ in (4.25), where $a_1 \neq -1, 0, 1$, $n \geq 2$, and the X_j have finite variance, then X_1 is normal ([4.22], p. 104).
3. Suppose X_1, X_2, X_3 are independent symmetric RVs with median 0 and CDF that are continuous at 0. If $W_1 = X_1/X_3$, $W_2 = X_2/X_3$ have joint characteristic function $\mathrm{E}[\exp(it_1W_1 + it_2W_2)] = \exp(t_1^2 + t_2^2)^{1/2}$, then the X_j are IID normal ([4.22], p. 104). (Here W_1, W_2 are identically distributed Cauchy RVs.)

Remarks. Some of the above results have parallel versions that characterize the gamma populations when the support is restricted to positive values or when $\mathcal{F}$ is the scale family distribution. Some provide characterizations of the Poisson distribution when the support is restricted to nonnegative integers.

4.6 Other Continuous Distributions

4.6.1 Uniform

As noted earlier, some characterizations of the exponential lead to similar results for the uniform. In addition, several characterizations based on the properties of order statistics and sums of IID RVs do exist. We have noted earlier [see (4.18)] that the maximum correlation for any two order statistics is attained only for the uniform distribution. Below are some other conditions that characterize the uniform distribution over an interval whose endpoints are chosen appropriately. (See [4.30], p. 282–285 for original references.)

1. $\mathrm{Cov}(X_{1:2}, X_{2:2}) = (1/3)\mathrm{Var}(X)$ or equivalently $\mathrm{E}(X_{2:2} - X) = [(1/3)\mathrm{Var}(X)]^{1/2}$.
2. $[\mathrm{E}(X_{2:2})]^2 = (4/3)\mathrm{E}(X^2)$. Here and above, the right side represents the maximum possible value for the left side expression.
3. $X_{2:2} - X_{1:2} \stackrel{d}{=} X_{1:2}$.
4. For some $n \geq 3$, $\mathrm{E}(X_1 | X_{1:n}, X_{n:n}) = (1/2)(X_{1:n} + X_{n:n})$ almost surely and F is continuous.

4.6.2 Gamma

The following conditions characterize the gamma distribution, whose PDF is given by

$$f(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} \exp(-x/\beta) x^{\alpha-1}, \; x > 0\,, \tag{4.28}$$

where $\alpha > 0$ and $\beta > 0$ are the shape and scale parameters, respectively ([4.22], p. 350–354; [4.5], p. 407–410).

1. X_1 and X_2 are independent nondegenerate RVs, and $X_1 + X_2$ and X_1/X_2 are independent.
2. For independent RVs $X_1, \ldots, X_n$, and $T = \sum_{j=1}^n X_j$, $\left(\frac{X_1}{T}, \ldots, \frac{X_n}{T}\right)$ and T are independent.
3. In a random sample of size n from a positive RV X for which $\mathrm{E}(1/X)$ is finite, the conditional expectation $\mathrm{E}(\sum_{j=1}^n X_j^{-1} | X_1 - \bar{X})$ is a constant.
4. The distribution in the scale family of distributions with support $(0, \infty)$ that has the smallest attainable Fisher information measure.
5. The distribution with the maximum entropy among distributions with support $(0, \infty)$ and have specified $\mathrm{E}(X)$ and $\mathrm{E}(\log(X))$.

4.6.3 Weibull

Some exponential characterizations easily lead to characterizations of the Weibull distribution with the PDF

$$f(x) = \frac{\alpha x^{\alpha-1}}{\beta^\alpha} \mathrm{e}^{-(x/\beta)^\alpha}, \; x > 0\,, \tag{4.29}$$

where $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter. This is also a distribution that is *min-stable* and is one of the extreme-value distributions to which the sample minima from random samples may converge. Further, each of the following properties characterize it ([4.22], Sect. 21.9).

1. X_1, X_2 are independent nonnegative RVs and for some $a, b \in (0, 1)$
$$\min(X_1, X_2) \stackrel{d}{=} aX_1 \stackrel{d}{=} bX_2\,. \tag{4.30}$$
2. $\mathrm{Var}(X_{k+1:n}^\alpha | X_{k:n} = x)$ is a constant.
3. $X_1, \ldots,$ are IID and N is an independent RV with support $\{2, 3, \ldots, \}$, and $N^{1/\alpha} X_{1:N} \stackrel{d}{=} X_1$.
4. In the scale family of distributions, the distribution for which the Fisher information in the right-censored sample $X_{1:n}, \ldots, X_{k:n}$ is the same as in a random sample of size k for all n and $k \leq n$ ([4.3], p. 226).

4.6.4 Gumbel and Other Extreme-Value Distributions

For an arbitrary parent distribution, if $X_{n:n}$ has a nondegenerate limiting distribution, possibly after suitable standardization, then it is known that the limiting CDF has one of the following forms, except for a change of location and scale:

$$\begin{aligned} \text{(Fréchet)} \quad G_1(x;\alpha) &= 0 \quad x \leq 0, \alpha > 0\,, \\ &= \exp(-x^{-\alpha}) \quad x > 0\,; \end{aligned} \tag{4.31}$$

$$\begin{aligned} \text{(Weibull)} \quad G_2(x;\alpha) &= \exp\left[-(-x)^\alpha\right] \\ &\quad x \leq 0, \alpha > 0\,, \\ &= 1 \quad x > 0\,; \end{aligned} \tag{4.32}$$

$$\begin{aligned} \text{(Gumbel)} \quad G_3(x) &= \exp\left(-\mathrm{e}^{-x}\right) \\ &\quad -\infty < x < \infty\,. \end{aligned} \tag{4.33}$$

The CDF $G_2(x;\alpha)$ above is that of $-X$ where X has the Weibull PDF given in (4.29). These distributions form the class of *max-stable* distributions characterized by the following result ([4.31], p. 38).

Theorem 4.3

Let F be a nondegenerate CDF and a_n and b_n be two sequences such that $(X_{n:n}-a_n)/b_n \stackrel{d}{=} X_1$ for integers n_1 and n_2 such that $b_{n_2} \neq 1$ and $\log(b_{n_1})/\log(b_{n_2})$ is irrational. Then there are constants a, $b>0$ and $\alpha>0$ such that $F(a+bx)$ is either (4.31) or (4.32). If $b_{n_2}=1$ and a_{n_1}/a_{n_2} is irrational, then $F(a+bx)$ is given by (4.33).

Additional results for the Gumbel distribution are available. Here are two characterizing properties.

1. $\mathrm{E}(R'_n - R'_{n+1}|R'_{n+1})$ is a constant for some n, where the R'_n is the nth lower record value.
2. $R'_n - R'_m$ and R'_m are independent for some $n<m$.

4.6.5 Pareto

The Pareto distribution, used as a model for income distributions and as the limit distribution of residual lifetime, has the CDF

$$F(x) = 1-\left\{1+[(x-\mu)/\sigma]^{1/\gamma}\right\}^{-\alpha}, \quad x \geq \mu, \sigma>0, \gamma>0, \alpha>0\,. \tag{4.34}$$

Several characterizations exist for this distribution, especially when $\gamma=1$, in which case the resulting distribution is called the Pareto distribution of the second kind (II); one obtains the Pareto distribution of the first kind (I) when, in addition, $\mu=\sigma$. (See [4.32], Sect. 3.7. or [4.22], Sect. 21.9.) Since $-\log[1-F(X)] = \alpha\log\{1+[(X-\mu)/\sigma]^{1/\gamma}\}$ is the standard exponential, its numerous characterizations provide simple counterparts for the Pareto distribution. For example, independence of exponential spacings is equivalent to the independence of the ratios of order statistics from a Pareto I distribution. The Pareto II (with $\alpha>1$) is the only distribution for which $h(x)m(x)$ is a constant where the failure rate $h(x)$ and the MRL function $m(x)$ are given by (4.5) and (4.7), respectively. Another characterization is that, if W is a continuous RV with support (0, 1) and is independent of X, and $\{WX|WX\geq\mu\} \stackrel{d}{=} X$, then X must be a Pareto II RV. Here W can be seen as the proportion *underreported* or *undamaged*. For some recent results on generalized Pareto distributions, see [4.33].

4.6.6 Inverse Gaussian (IG)

This distribution has the PDF

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}}\exp\left(-\frac{\lambda}{2\nu^2}\frac{(x-\nu)^2}{x}\right), \quad x>0\,, \tag{4.35}$$

where the parameters are $\nu>0$ and $\lambda>0$. It arises as the waiting time to cross a certain threshold in Brownian motion. Its characterizations often mimic those for the normal distribution. For example, the IG distribution has the maximum entropy subject to certain restrictions on $\mathrm{E}(X)$ and $\mathrm{E}(1/X)$. For a random sample $X_1,\ldots,X_n$, let $Y=(1/n)\sum_{i=1}^n X_i^{-1} - \bar{X}^{-1}$. Then the population is IG if either $\bar{X}$ and Y are independent, or the regression $\mathrm{E}(Y|\bar{X})$ is a constant ([4.34], Chapt. 3).

4.7 Poisson Distribution and Process

The Poisson distribution, commonly known through its PDF

$$\Pr(X=j) = \mathrm{e}^{-\lambda}\frac{\lambda^j}{j!}, \quad j=0,1,2,\ldots;\lambda>0\,, \tag{4.36}$$

appears often in the engineering literature as a model for rare events and in queueing or reliability studies. We write X is Poi(λ) if (4.36) holds. The earliest result seems to be due to *Raikov* (1938) ([4.35], Sect. 4.8) who showed that, if X_1 and X_2 are independent and X_1+X_2 is Poisson, then each of them should be Poisson RVs. It is known that, if X_1 is Poi(λ_1) and X_2 is Poi(λ_2) and they are independent, the conditional distribution of X_1 given $X_1+X_2=n$ is Bin[n, $\lambda_1/(\lambda_1+\lambda_2)$], i. e., binomial with n trials and success probability $p=\lambda_1/(\lambda_1+\lambda_2)$. This property has led to various characterizations of the Poisson distribution. For example, if the conditional distribution is binomial, then X_1 and X_2 are both Poisson RVs.

One particularly interesting set up that identifies the Poisson distribution is the *damage model* due to *Rao*. The associated characterization result due to *Rao* and *Rubin* is the following ([4.7], p. 164):

Theorem 4.4

Let X and Y be nonnegative integer-valued RVs such that $\Pr\{X=0\}<1$ and, given $X=n$, Y is Bin(n, p) for each $n \geq 0$ and a fixed $p \in (0,1)$. Then the Rao–Rubin condition, given by

$$\Pr(Y=j)=\Pr(Y=j|Y=X), \quad j=0,1,\ldots, \tag{4.37}$$

holds iff X is Poisson.

The condition (4.37) is equivalent to the condition

$$\Pr(Y=j|Y=X)=\Pr(Y=j|Y<X) \tag{4.38}$$

which can be interpreted as follows. Suppose X is the number of original counts and Y is the number actually available, the remaining being lost due to damage according to the binomial model. Then if the probability distribution of the actual counts remains the same whether damage has taken place or not, the number of original counts must be Poisson. Incidentally, the number of observations that survived is also Poisson. The above damage model can be seen as binomial *splitting* or *thinning* and a similar notion is that of binomial *expanding*. It also leads to a characterization of the Poisson distribution.

A weaker version of (4.37) can be used to characterize the Poisson distribution by restricting the family $\mathcal{F}$ under consideration. Let X belong to the family of the power-series distribution, i. e., it has the PDF

$$\Pr(X=j)=\frac{a_j\theta^j}{A(\theta)}, \quad j=0,1,\ldots \tag{4.39}$$

Suppose that given $X=n$, Y has support $0,\ldots,n$, and has mean np and variance $np(1-p)$, where p does not depend on θ. Then $\mathrm{E}(Y|Y=X)=\mathrm{E}(Y)$ and $\mathrm{Var}(Y|Y=X)=\mathrm{Var}(Y)$ iff X is Poisson.

Poisson characterizations based on the properties of the sample mean $\bar{X}$ and variance S^2 from a random sample are known. In the power-series family (4.39), if $\mathrm{E}(S^2|\bar{X}>0)=1$, then the population is necessarily Poisson. When X is assumed to be nonnegative, if $\mathrm{E}(S^2|\bar{X})=\bar{X}$, the parent is Poisson also. See [4.35], Sect. 4.8, for relevant references. Characterizations based on the discrete analogue of the Skitovich–Darmois theorem (Theorem 4.2) are available [4.36]. It is also known that, in a wide class of distributions on the set of integers, the Poisson distribution is characterized by the equality sign in a discrete version of the Stam inequality for the Fisher information; the continuous version yields a normal characterization [4.37]. Another *normal-like* result is the Poisson characterization by the identity $\mathrm{E}(X)\mathrm{E}[g(X+1)]=\mathrm{E}[Xg(X)]$ assumed to hold for every bounded function $g(.)$ on the integers [4.38].

Poisson Process

A renewal process is a counting process $\{N(t), t \geq 0\}$ where the inter-arrival times of events are IID with CDF F. The (homogeneous) Poisson process is characterized by the fact that F is exponential. Several characterizations of a Poisson process in the family of renewal processes do exist. For example, if a renewal process is obtained by the superposition of two independent renewal processes, then the processes must be Poissonian. Several are tied to the exponential characterizations from random samples. Other characterizations of interest are based on the properties of the *current age* and *residual lifetime* distributions.

Let X_i represent the IID inter-arrival times and $S_n=X_1+\cdots+X_n$, so that $N(t)=\sup(m: S_m \leq t)$. Then $A(t)=t-S_{N(t)}$ represents the current age or *backward recurrence time* at t and $W(t)=S_{N(t)+1}-t$ is the residual lifetime or *forward recurrence time* at t, $t \geq 0$. A good summary of the available results is provided in [4.39], p. 674–684. Chapter 4 of [4.31] contains an early account of various characterizations of the Poisson process that include thinned renewal processes and geometric compounding. We state below a few simple characterizing properties of the Poisson process:

1. Either $\mathrm{E}[W(t)]$ or $\mathrm{Var}[W(t)]$ is a finite constant for all $t>0$.
2. F is continuous with $F^{-1}(0)=0$, and for some fixed t, $A(t)$ and $W(t)$ are independent.
3. F is continuous and $\mathrm{E}[A(t)|N(t)=n]=\mathrm{E}[X_1|N(t)=n]$ for all $t>0$ and all $n \geq 1$.
4. F is continuous and $\mathrm{E}[A(t)]=\mathrm{E}[\min(X_1,t)]$ for all $t>0$.

4.8 Other Discrete Distributions

4.8.1 Geometric

Numerous versions of the LMP of the geometric distribution have led to several characterizations of the geometric distribution with PDF

$$\Pr(X = j) = (1-p)^j p,\ j = 0, 1, \ldots \tag{4.40}$$

Here the LMP means $\Pr(X > x + j | X \geq x) = \Pr(X > j)$, $j, x = 0, 1, \ldots$ When X has the above PDF, the following properties hold:

1. $\mathrm{E}(X - x | X \geq x) = \mathrm{E}(X)$, $x = 0, 1, \ldots$.
2. $|X_1 - X_2| \stackrel{d}{=} X$.
3. $\Pr(X_{1:n} \geq 1) = \Pr(X_1 \geq n)$, $n \geq 1$.
4. $X_{j+1:n} - X_{j:n} \stackrel{d}{=} X_{1:n-j}$, $1 \leq j < n$.
5. $(X_{k:n} - X_{j:n} | X_{j+1:n} - X_{j:n} > 0) \stackrel{d}{=} 1 + X_{k-j:n-j}$, $1 \leq j < k \leq n$.
6. $X_{1:n}$ and $X_{j:n} - X_{1:n}$ are independent.

Each of these is shown to be a characteristic property of the geometric or slightly modified versions of that distribution, under mild conditions [4.40].

In terms of the upper record values, the following properties hold for the geometric parent and characterize it ([4.4], Sect. 4.6).

1. $R_1, R_2 - R_1, R_3 - R_2, \ldots$ are independent.
2. $E(R_{n+1} - R_n \mid R_n)$, $E(R_{n+2} - R_{n+1} \mid R_n)$, and $E[(R_2 - R_1)^2 \mid R_1]$ are constants.
3. $R_{n+1} - R_n \stackrel{d}{=} R_1$, $n \geq 1$.

4.8.2 Binomial and Negative Binomial

The damage model, discussed in Theorem 4.4, also produces a characterization of the binomial distribution in that, if (4.37) holds and X is Poisson, then the damage process is binomial. Another characterization of the binomial distribution assumes that the RVs X and Y are independent, and that the conditional distribution of X given $X + Y$ is hypergeometric [4.41]. When the conditional distribution is negative hypergeometric, a similar result for the negative binomial distribution is obtained.

Remarks. Characterizations of other discrete distributions are limited. For results on hypergeometric and logarithmic distributions, see [4.35]. Characterizations of discrete distributions based on order statistics are discussed in [4.40]. See [4.42] for characterizations based on weighted distributions when $\mathcal{F}$ is the power-series family in (4.39).

4.9 Multivariate Distributions and Conditional Specification

Characterization results are less common for multivariate distributions. Notable exceptions are the multivariate normal and the Marshall–Olkin multivariate exponential distribution. First we discuss another dimension to multivariate characterizations, namely the specification of the properties of the conditional distribution(s). For example, can one identify the joint PDF $f(x, y)$ using the properties of the conditional PDFs $f(x|y)$ and $f(y|x)$? This has been an active area of research in recent years. See [4.43] for an excellent account of the progress. We present one such result.

Theorem 4.5

Let $f(x, y)$ be a bivariate PDF where conditional PDFs belong to natural parameter exponential families with full rank given by

$$f(x|y) = r_1(x)\beta_1[\boldsymbol{\theta}_1(y)] \exp\left[\boldsymbol{\theta}_1(y)'\boldsymbol{q}_1(x)\right] \tag{4.41}$$

and

$$f(y|x) = r_1(y)\beta_2[\boldsymbol{\theta}_2(x)] \exp\left[\boldsymbol{\theta}_2(x)'\boldsymbol{q}_2(y)\right] \tag{4.42}$$

where $\boldsymbol{\theta}_1(y)$ and $\boldsymbol{q}_1(x)$ are $k_1 \times 1$ vectors, and $\boldsymbol{\theta}_2(x)$ and $\boldsymbol{q}_1(y)$ are $k_2 \times 1$ vectors, and the components of $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are linearly independent. Then the joint PDF is of the form

$$f(x, y) = r_1(x)r_2(y) \exp[\boldsymbol{A}(x, y)] \tag{4.43}$$

where $\boldsymbol{A}(x, y) = [1, \boldsymbol{q}_1(x)']\boldsymbol{M}[1, \boldsymbol{q}_2(y)']'$ for a suitable matrix $\boldsymbol{M} = (m_{ij})$, whose elements are chosen so that $f(x, y)$ integrates to 1.

When both the conditional distributions are normal, this result implies that

$$f(x, y) \propto \exp[(1, x, x^2)\boldsymbol{M}(1, y, y^2)'] \tag{4.44}$$

and the classical bivariate normal corresponds to the condition $m_{23} = m_{32} = m_{33} = 0$ [4.44].

Instead of the conditional PDF, the conditional distribution may be specified using regression functions, say $\mathrm{E}(Y|X=x)$. Then the joint distribution can be determined in some cases. For example, suppose X given $Y=y$ is $\mathrm{N}(\alpha y, 1)$, i. e., normal with mean αy and unit variance, and $\mathrm{E}(Y|X=x)=\beta x$. Then $0<\alpha\beta<1$ and (X, Y) is bivariate normal [4.44].

The conditional specification could be in terms of the conditional SF $\Pr(Y>y|X>x)$, or in the form of the marginal distribution of X and the conditional distribution of X given $Y=y$. Sometimes these together can also identify the joint distribution.

4.9.1 Bivariate and Multivariate Exponential Distributions

Marshall and *Olkin* [4.45] introduced a bivariate exponential (BVE) distribution to model the component lifetimes in the context of a shock model. Its (bivariate) SF, with parameters $\lambda_1>0$, $\lambda_2>0$, and $\lambda_{12}\geq 0$, is given by

$$\Pr(X>x, Y>y)=\mathrm{e}^{-\lambda_1 x-\lambda_2 y-\lambda_{12}\max(x,y)}, \quad x, y>0\,. \tag{4.45}$$

Here X is $\exp(\lambda_1+\lambda_{12})$ and Y is $\exp(\lambda_2+\lambda_{12})$. The joint distribution is characterized by the following conditions: (a) X and Y are marginally exponential. (b) $\min(X, Y)$ is exponential, and (c) $\min(X, Y)$ and $|X-Y|$ are independent.

The LMP in (4.18) that characterized the univariate exponential can be extended as

$$\Pr(X>x+t_1, Y>y+t_2|X>x, Y>y) = \Pr(X>t_1, Y>t_2)\,. \tag{4.46}$$

If (4.46) is assumed to hold for all $x, y, t_1, t_2\geq 0$, then X and Y are necessarily independent exponential RVs. The SF (4.45) would satisfy (4.46) for all $x, y\geq 0$ and $t_1=t_2=t\geq 0$. This condition, often referred to as *bivariate* LMP, is equivalent to assuming that both (b) and (c) above hold. While the Marshall–Olkin BVE distribution has exponential marginals and bivariate LMP, it is not absolutely continuous. If one imposes the LMP and absolute continuity, the marginal distributions will no longer be exponential [4.46].

There are other multivariate distributions that are characterized by the multivariate versions of the failure-rate function (see [4.47], p. 403–407).

4.9.2 Multivariate Normal

An early characterization of the classical multivariate normal (MVN) random vector, known as Cramér–Wold Theorem, is that every linear combination of its components is univariate normal. Most of the characterizations of the univariate normal distribution discussed in Sect. 4.5 easily generalize to the MVN distribution. For example, the independence of nonsingular transforms of independent random vectors [see (4.25) for the univariate version], independence of the sample mean vector and sample covariance matrix, maximum entropy with a given mean vector and covariance matrix, are all characteristic properties of the MVN distribution. There are, of course, results based on conditional specifications. We mention two.

For an m-dimensional RV $\boldsymbol{X}$, let $\boldsymbol{X}_{(i,j)}$ be the vector $\boldsymbol{X}$ with coordinates i and j deleted. If, for each i, j the conditional distribution of (X_i, X_j) given $\boldsymbol{X}_{(i,j)}=\boldsymbol{x}_{(i,j)}$ is BVN for each $\boldsymbol{x}_{(i,j)}$, then $\boldsymbol{X}$ is MVN ([4.43], p. 188).

If $X_1, \cdots, X_m$ are jointly distributed RVs such that $(X_1, \cdots, X_{m-1})\stackrel{d}{=}(X_2, \cdots, X_m)$, and X_m given $\{X_1=x_1, X_2=x_2, \cdots, X_{m-1}=x_{m-1}\}$ is $\mathrm{N}\left(\alpha+\sum_{j=1}^{m-1}\beta_j x_j, \sigma^2\right)$, then $(X_1, \cdots, X_m)$ are jointly m-variate normal ([4.47], p. 157).

Excellent summaries of characterizations of the bivariate and multivariate normal distributions are available, respectively, in Sect. 46.5 and Sect. 45.7 of [4.47] (see also, the review [4.48]).

4.9.3 Other Distributions

Characterization results for other multivariate distributions are not common. A few characterizations of the *multinomial distribution* are available ([4.49], Sect. 35.7), and these are natural extensions of the binomial characterizations. One result is that, if the sum of two independent vectors is multinomial, then each is multinomial. There are also a few characterizations of the Dirichlet distribution, a multivariate extension of the beta distribution over $(0, 1)$ ([4.47], Sect. 49.5). It has the characteristic property of *neutrality*, which can be described for $m=2$ components as follows. For two continuous RVs X and Y such that $X, Y\geq 0$ and $X+Y\leq 1$, neutrality means X and $Y/(1-X)$ are independent, and Y and $X/(1-Y)$ are independent.

The multivariate Pareto distribution due to Mardia, having the multivariate SF

$$\Pr(X_1 > x_1, \ldots, X_m > x_m) = \left[1 + \sum_{i=1}^{m} \left(\frac{x_i}{\sigma_i}\right)\right]^{-\alpha},$$
$$x_1, \ldots, x_m > 0\,, \tag{4.47}$$

accepts characterizations that are based on conditional specifications. Marginally, the X_i here are Pareto II RVs.

A few papers that characterize bivariate distributions with geometric marginals do exist. Some are related to the Marshall–Olkin BVE.

4.10 Stability of Characterizations

Consider the LMP in (4.18) that characterizes the exponential distribution. Now suppose the LMP holds approximately in the sense

$$\sup_{x\geq 0, y\geq 0} \left|\Pr(X > x+y | X > x) - \Pr(X > y)\right| \leq \epsilon\,. \tag{4.48}$$

The question of interest is how close the parent CDF F is to an exponential CDF. It is known ([4.21], p. 7) that, when X is nondegenerate and $F^{-1}(0) = 0$, if (4.48) holds then $\mathrm{E}(X)$ is finite and, with $\mathrm{E}(X) = 1/\lambda$,

$$\sup_{x\geq 0} \left|\Pr(X > x) - \exp(-\lambda x)\right| \leq 2\epsilon\,. \tag{4.49}$$

This result provides an idea about the stability of the LMP of the exponential distribution. There are many such results—mostly for the exponential and normal distributions. Such results involve appropriate choices of metrics for measuring the distance between: (a) the characterizing condition and the associated perturbation, and (b) the CDF being characterized and the CDF associated with the perturbed condition. We will mention below a few simple stability theorems. It is helpful to introduce one popular metric measuring the distance between two distributions with associated RVs X and Y:

$$\rho(X, Y) \equiv \sup_{-\infty < x < \infty} |\Pr(X \leq x) - \Pr(Y \leq x)|\,. \tag{4.50}$$

Note that (4.49) basically says that $\rho(X, Y) \leq 2\epsilon$, where Y is an $\exp(\lambda)$ RV.

For the exponential parent, the constancy of $\mathrm{E}[(X_{k+1:n} - X_{k:n})|X_{k:n}]$ is a characterizing property. The associated stability result is the following [4.22, p. 545], [4.50]:

If F has support $(0, \infty)$ and is strictly increasing in its domain, and $\gamma > 0$ is such that

$$\left|\mathrm{E}\left[(X_{k+1:n} - X_{k:n})\middle| X_{k:n} = x\right] - \gamma\right| \leq \epsilon\,,$$
$$\text{for almost all } x \geq 0\,, \tag{4.51}$$

then, there exist positive constants λ_1, λ_2 and c that depend only on γ and $n-k$ such that $\rho(X, Y) \leq c\epsilon^{1/(n-k)} \exp(-\lambda_2 x)$, where Y is $\exp(\lambda_1)$.

For the normal distribution, we state stability results for two classical characterizations.

1. *Pólya's characterization* [4.5, p. 298]. Let X_1 and X_2 be IID with zero mean, unit variance and $\mathrm{E}(|X_1|^3) \leq M < \infty$. Let $Y = (X_1 + X_2)/\sqrt{2}$. If $\rho(X_1, Y) \leq \epsilon$, then $\rho(X_1, Z) \leq c\epsilon^{1/3}$, where c depends only on M and Z is standard normal. (See the recent work [4.51] for another metric of comparison.)
2. *Independence of $\bar{X}$ and S^2* [4.52]. Suppose that for all x, y,

$$\left|\Pr(\bar{X} \leq x, S^2 \leq y) - \Pr(\bar{X} \leq x)\Pr(S^2 \leq y)\right| \leq \epsilon\,. \tag{4.52}$$

Then there exists a (possibly degenerate) $\mathrm{N}(\mu, \sigma^2)$ RV Y such that $\rho(X, Y) \leq c[\log(1/\epsilon)]^{-1/2}$, where c depends only on n, and ρ is given by (4.50).

Many such results are available in the several monographs on stability of characterizations edited by *Kalashnikov* and *Zolotarov* and others (see e.g., [4.52, 53]). These have come from the periodic conferences held in Eastern Europe.

4.11 Applications

A characterization can be of use in the construction of goodness-of-fit tests and in the examination of the consequences of the modeling assumptions. It can be helpful in some simulation studies. A full characterization states that a condition C on the sample data is necessary and sufficient for the condition $F \in \mathcal{F}_0$ to hold. The *neces-*

sity part can be used for the simulation study of the properties of the sample statistics involved in the condition C. For example, if one is interested in the properties of a function of the random sample mean $\bar{X}$ and variance S^2 from a standard normal population, one can start with two *independent* statistics, one normal and the other a χ^2. We can simulate these statistics directly, and study the empirical properties of the function of interest. The characterization of the exponential based on the independence and exponentially distributed nature of spacings of order statistics, can be used to simulate all or a selected set of exponential order statistics without any sorting.

The *sufficiency* part can be used for checking implications of modeling assumptions or of their compatibility. Again consider the situation where the researcher is willing to accept, based on past data, say, that $\bar{X}$ and variance S^2 are independent. This is equivalent to assuming that the sample is from a normal population. As another example, consider the early characterization result, proved by *Cramér* in 1937, which states that, if X and Y are independent and the sum is assumed to be normal, then each of them must be normal (see [4.24], p. 53). Thus, if a researcher is willing to assume the independence of these RVs and that the sum is normal, this is the same as assuming that both X and Y are individually normal and are independent. This implication can be gainfully employed to infer that $aX+bY$ is $N(a\mu_X+b\mu_Y, a^2\sigma_X^2+b^2\sigma_Y^2)$.

Characterizing properties, when used as necessary and sufficient conditions, naturally lead to both informal (graphical) and formal goodness-of-fit tests. The various plots such as the quantile–quantile (Q–Q) plots, hazard function plots, or the MRL plots are all based on the characterizing properties of these functions. For example, in Q–Q plots, the sample quantile $X_{j:n}$ is plotted against the corresponding hypothesized population quantile $F_0^{-1}(p_j)$, where F_0 is the standardized form of F, and p_j is sometimes chosen as $(j-(1/2))/n$. (See [4.3], p. 270 for other choices for p_j.) If a linear fit is unreasonable, then one may infer that the assumption that F is the parent CDF is untenable. Probability plots (see [4.54], Chapt. 3), and the hazard-function plots ([4.54], Chapt. 4), popular in reliability studies, also provide similar informal checks of the fit of the assumed distribution. MRL plots, where $\frac{1}{k}\sum_{j=n-k+1}^{n}(x_{j:n}-x_{n-k:n})$ is plotted against the $x_{n-k:n}$ (see, [4.55], p. 296), are used to determine the appropriate domain of attraction for the extreme order statistics.

In theory, a formal goodness-of-fit test can be constructed from any characterization. For example, the independence of spacings of order statistics of a random sample from a continuous distribution implies the CDF is exponential, and thus can be used to construct a goodness-of-fit test even with a (type II) censored sample. Such a test does have good power properties. Another example is the test based on the maximum-entropy property of the normal distribution [4.56]. See [4.57–59] for some goodness-of-fit tests inspired by characterizations. A nice overview is provided in [4.60]. However, not all characterizations lead to powerful tests. Also, creation of goodness-of-fit tests that exploit the concerned characterizing property to the full extent may not be easy. Perhaps this explains the sparsity of such applications of characterizations.

4.12 General Resources

The literature on characterization is extensive. Over 1000 papers have appeared to date; however, several review articles, monographs (e.g., [4.5, 7, 31, 43]) and encyclopedic books on distributions by *Johnson*, *Kotz*, and their coauthors ([4.22, 30, 35, 47, 49]) have served as excellent filters. Other sources include the volume [4.61], which contains papers on characterizations presented at the Calgary conference in 1974, and the monograph [4.62], which treats characterization (identification) in a broader context. For results based on order statistics see [4.3, Sect. 6.7]; see also [4.63].

Here, we have given an informal introduction to characterizations using a small fraction of the available results. For economy, we have cited the above general and other (distribution-specific) secondary resources in this survey. To get a full appreciation of this active area of research, one should consult many of the primary as well as secondary sources.

References

4.1 C. R. Rao: *Linear Statistical Inference and Its Applications*, 2nd edn. (Wiley, New York 1973)

4.2 R. E. Barlow, F. Proschan: *Statistical Theory of Reliability and Life Testing: Probability Models* (To Begin With, Silverspring 1981)

4.3 H. A. David, H. N. Nagaraja: *Order Statistics*, 3rd edn. (Wiley, New York 2003)

4.4 B. C. Arnold, N. Balakrishnan, H. N. Nagaraja: *Records* (Wiley, New York 1998)

4.5 A. M. Kagan, Yu. V. Linnik, C. R. Rao: *Characterization Problems in Mathematical Statistics* (Wiley, New York 1973)

4.6 B. Ramachandran, K.-S. Lau: *Functional Equations in Probability Theory* (Academic, Boston 1991)

4.7 C. R. Rao, D. N. Shanbhag: *Choquet-Deny Type Functional Equations with Applications to Stochastic Models* (Wiley, Chichester 1994)

4.8 K.-S. Lau, C. R. Rao: Integrated Cauchy functional equations, characterizations of the exponential law, Sankhyā Ser. A **44**, 72–90 (1982)

4.9 C. R. Rao, D. N. Shanbhag: Extensions of a characterization of an exponential distribution based on a censored ordered sample. In: *Advances in the Theory and Practice of Statistics*, ed. by N. L. Johnson, N. Balakrishnan (Wiley, New York 1997) pp. 431–440

4.10 U. Kamps: Characterizations of distributions by recurrence relations and identities for moments of order statistics. In: *Order Statistics: Theory and Methods*, Handbook of Statistics, Vol. 16, ed. by N. Balakrishnan, C. R. Rao (North-Holland, Amsterdam 1998) pp. 291–311

4.11 A. V. Kakosyan, L. B. Klebanov, J. A. Melamed: *Characterization of distributions by the method of intensively monotone operators*, Lect. Notes in Math., Vol. 1088 (Springer, Berlin Heidelberg New York 1984)

4.12 J. Galambos: Characterizations of distributions. In: *Encyclopedia of Statistical Sciences*, Vol. 1, ed. by S. Kotz, N. L. Johnson (Wiley, New York 1982) pp. 422–428

4.13 J. A. Shohat, J. D. Tamarkin: *The Problem of Moments*, Math. Surveys, Vol. 1 (American Mathematical Society, New York 1943)

4.14 J. S. Huang: Moment problem of order statistics: a review, Int. Stat. Rev. **57**, 59–66 (1989)

4.15 M. Franco, J. M. Ruiz: Characterization based on conditional expectations of adjacent order statistics: a unified approach, Proc. Am. Math. Soc. **127**, 861–874 (1999)

4.16 N. A. Langberg, R. V. León, F. Proschan: Characterization of nonparametric classes of life distributions, Ann. Probab. **8**, 1163–1170 (1980)

4.17 L. Bondesson: A generalization of Poincaré's characterization of exponential families, J. Stat. Plan. Infer. **63**, 147–155 (1997)

4.18 B. C. Arnold, J. S. Huang: Characterizations. In: *The Exponential Distribution: Theory, Methods and Applications*, Lect. Notes Math., ed. by N. Balakrishnan, A. P. Basu (Gordon Breach, Amsterdam 1995) pp. 185–203

4.19 J.-L. Xu: Characterizations of the exponential distribution via the blocking time in a queueing system with an unreliable server, J. Appl. Probab. **35**, 236–239 (1998)

4.20 U. Gather, U. Kamps, N. Schweitzer: Characterizations of distributions via identically distributed functions of order statistics. In: *Order Statistics: Theory and Methods*, Handbook of Statistics, Vol. 16, ed. by N. Balakrishnan, C. R. Rao (North-Holland, Amsterdam 1998) pp. 257–290

4.21 T. Azlarov, N. A. Volodin: *Characterization Problems Associated With the Exponential Distribution* (Springer, Berlin Heidelberg New York 1986)

4.22 N. L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions*, Vol. 1 (Wiley, New York 1994)

4.23 T. S. Ferguson: On a Rao–Shanbhag characterization of the exponential/geometric distribution, Sankhyā A **64**, 246–255 (2002)

4.24 H. Cramér: *Random Variables and Probability Distributions*, 3rd edn. (Cambridge Univ. Press, Cambridge 1970)

4.25 E. Lukacs, R. G. Laha: *Applications of Characteristic Functions* (Griffin, London 1964)

4.26 A. M. Mathai, G. Pederzoli: *Characterizations of the Normal Probability Law* (Halsted, New York 1977)

4.27 W. Bryc: *The Normal Distribution. Characterizations with Applications*, Lect. Notes Stat., Vol. 100 (Springer, Berlin Heidelberg New York 1995)

4.28 L. Bondesson: When is the sample mean BLUE?, Scand. J. Stat. **3**, 3116–120 (1976)

4.29 P. K. Goel, M. H. DeGroot: Only normal distributions have linear posterior expectations in linear regression, J. Am. Stat. Assoc. **75**, 895–900 (1980)

4.30 N. L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions*, Vol. 2, 2nd edn. (Wiley, New York 1995)

4.31 J. Galambos, S. Kotz: *Characterizations of Probability Distributions*, Lect. Notes Math., Vol. 675 (Springer, Berlin Heidelberg New York 1978)

4.32 B. C. Arnold: *Pareto Distributions* (International Cooperative, Burtonsville 1983)

4.33 M. Asadi, C. R. Rao, D. N. Shanbhag: Some unified characterization results on generalized Pareto distributions, J. Stat. Plan. Infer. **93**, 29–50 (2001)

4.34 V. Seshadri: *The Inverse Gaussian Distribution. A Case Study in Exponential Families* (Oxford Univ. Press, Oxford 1993)

4.35 N. L. Johnson, S. Kotz, A. W. Kemp: *Univariate Discrete Distributions*, 2nd edn. (Wiley, New York 1992)

4.36 E. McKenzie: Linear characterizations of the Poisson distribution, Statist. Probab. Lett. **11**, 459–461 (1991)

4.37 A. Kagan: A discrete version of the Stam inequality, a characterization of the Poisson distribution, J. Stat. Plan. Infer. **92**, 7–12 (2001)

4.38 P. Diaconis, S. Zabell: Closed form summation for classical distributions: variations on a theme of de Moivre, Stat. Sci. **6**, 284–302 (1991)

4.39 B. L. S. Prakasa Rao: Characterization, identifiability for stochastic processes. In: *Stochastic Processes: Theory and Methods*, Handbook of Statistics, Vol. 19, ed. by D. N. Shanbhag, C. R. Rao (Elsevier, Amsterdam 2001) pp. 643–691

4.40 H. N. Nagaraja: Order statistics from discrete distributions (with discussion), Statistics **23**, 189–216 (1992)

4.41 P. Patil, V. Seshadri: Characterization theorems for some univariate probability distributions, J. Roy. Statist. Soc. Ser. B **26**, 286–292 (1964)

4.42 A. G. Pakes, J. Navarro, J. M. Ruiz, Y. del Aguila: Characterizations using weighted distributions, J. Stat. Plan. Infer. **116**, 389–420 (2003)

4.43 B. C. Arnold, E. Castillo, J. M. Sarabia: *Conditional Specification of Statistical Models* (Springer, Berlin Heidelberg New York 1999)

4.44 B. C. Arnold: Characterizations involving conditional specification, J. Stat. Plan. Infer. **63**, 117–131 (1997)

4.45 A. W. Marshall, I. Olkin: A multivariate exponential distribution, J. Am. Stat. Assoc. **62**, 30–44 (1967)

4.46 H. W. Block: A characterization of a bivariate exponential distribution, Ann. Stat. **5**, 808–812 (1977)

4.47 S. Kotz, N. Balakrishnan, N. L. Johnson: *Continuous Multivariate Distributions*, Vol. 1, 2nd edn. (Wiley, New York 2000)

4.48 G. G. Hamedani: Bivariate, multivariate normal characterizations: a brief survey, Commun. Stat. Theory Methods **21**, 2665–2688 (1992)

4.49 N. L. Johnson, S. Kotz, N. Balakrishnan: *Discrete Multivariate Distributions* (Wiley, New York 1997)

4.50 R. V. Yanushkevichius: *Stability of Characterizations of Probability Distributions* (Mokslas, Vilnius 1991) In Russian

4.51 R. Yanushkevichius: Stability of the characterization of normal distribution in the Laha–Lukacs theorem, Stat. Probab. Lett. **49**, 225–233 (2000)

4.52 R. Yanushkevichius: On ϵ-independence of sample mean, sample variance. In: *Stability Problems for Stochastic Models*, Lect. Notes Math., ed. by V. V. Kalashnikov, B. Penkov, V. M. Zolotarev (Springer, Berlin Heidelberg New York 1978) pp. 207–223

4.53 V. V. Kalashnikov, V. M. Zolotarev (eds.): *Stability Problems for Stochastic Models*, Lect. Notes in Math., Vol. 982 (Springer, Berlin Heidelberg New York 1983)

4.54 W. Nelson: *Applied Life Data Analysis* (Wiley, New York 1982)

4.55 P. Embrechts, C. Klüppelberg, T. Mikosch: *Modelling Extremal Events for Insurance and Finance* (Springer, Berlin Heidelberg New York 1997)

4.56 O. Vasicek: A test for normality based on sample entropy, J. R. Stat. Soc. Ser. B **38**, 54–59 (1976)

4.57 M. Csörgő, V. Seshadri, M. Yalovsky: Applications of characterizations in the area of goodnees of fit. In: *Statistical Distribution in Scientific Work*, Vol. 2, ed. by G. P. Patil, S. Kotz, J. K. Ord (Reidel, Dordrecht 1975) pp. 79–90

4.58 F. J. O'Reilly, M. A. Stephens: Characterizations, goodness of fit tests, J. R. Stat. Soc. Ser. B **44**, 353–360 (1982)

4.59 K. W. Morris, D. Szynal: Tests for uniformity, exponentiality using a characterization, J. Math. Sci. **111**, 3572–3581 (2002)

4.60 C. E. Marchetti, G. S. Mudholkar: Characterization theorems and goodness-of-fit tests. In: *Goodness-of-Fit Tests and Model Validity*, ed. by C. Huber-Carol, N. Balakrishnan, M. S. Nikulin, M. Mesbah (Birkhäuser, Boston 2002) pp. 125–142

4.61 G. P. Patil, S. Kotz, J. K. Ord (eds.): A Modern Course on Statistical Distributions in Scientific Work. In: *Characterizations and Applications*, Vol. 3 (Reidel, Dordrecht-Boston 1975)

4.62 B. L. S. Prakasa Rao: *Identifiability in Stochastic Models – Characterization of Probability Distributions* (Academic, Boston 1992)

4.63 C. R. Rao, D. N. Shanbhag: Recent approaches to characterizations based on order statistics and record values. In: *Order Statistics: Theory and Methods*, Handbook of Statistics, Vol. 16, ed. by N. Balakrishnan, C. R. Rao (North-Holland, Amsterdam 1998) pp. 231–256

5. Two-Dimensional Failure Modeling

For many products (for example, automobiles), failures depend on age and usage and, in this case, failures are random points in a two-dimensional plane with the two axes representing age and usage. In contrast to the one-dimensional case (where failures are random points along the time axis) the modeling of two-dimensional failures has received very little attention. In this chapter we discuss various issues (such as modeling process, parameter estimation, model analysis) for the two-dimensional case and compare it with the one-dimensional case.

All products are unreliable in the sense that they degrade and fail with age and/or usage, and ultimately fail. Reliability of a product conveys the concept of dependability and the absence of failures. Reliability theory deals with various aspects of product reliability and encompasses various reliability issues. These include the following:

1. Reliability science to understand the degradation leading to failures.
2. Reliability engineering to design and manufacture reliable products.
3. Reliability management to manage the activities during the design and manufacture of products and the operation of unreliable products.
4. Reliability modeling to build models to obtain solutions to a variety of reliability-related problems in predicting, estimating, and optimizing the performance of unreliable products, the impact of unreliability and actions to mitigate the impact.

The modeling of failures is an important element of reliability modeling. In one-dimensional failure modeling, failures are random points along a one-dimensional axis representing age or usage. For many products (for example, automobiles), failures depend on age and usage and, in this case, failures are random points in a two-dimensional plane with the two axes representing age and usage. Models play an important role in decision-making. Many different types of models are used and these can be found in [5.1–4]. One-dimensional modeling has received considerable attention and so there is a vast literature covering this area. In contrast, two-dimensional failure modeling has received relatively little attention. In this chapter we discuss various issues relating to two-dimensional failure modeling.

The outline of the chapter is as follows. In Sect. 5.1 we discuss various issues relating to the modeling of failures. Section 5.2 deals with the black-box approach (or empirical modeling) where the modeling is based

on failure and operational data. One-dimensional modeling based on the black-box approach is reviewed in Sect. 5.3, where we discuss the different issues and illustrate through an example involving a real case. This sets the background for the two-dimensional modeling discussed in Sect. 5.4. Two different approaches have been proposed for two-dimensional failure modeling. We discuss these two approaches and indicate topics for future research. In Sect. 5.5 we discuss a new approach and this is illustrated using the case discussed earlier. Finally, we conclude with some comments and remarks in Sect. 5.6.

5.1 Modeling Failures

In this section we briefly discuss various issues of importance in the modeling of failures.

5.1.1 Product Failures

Products can vary from simple (such as an electric kettle) to complex (such as an aeroplane). A product can be viewed as a system consisting of several parts and can be decomposed into a hierarchy of levels with the system at the top level and components at the lowest level and several levels (such as sub-system, assembly, sub-assembly and so on) in between. The failure of a product is due to the failure of one or more of its components.

The occurrence of failure depends on several factors. These include decisions made during the design and manufacture of the product, usage intensity and operating environment, and the maintenance actions (corrective and preventive) carried out during the operating life.

The modeling of failures can be done at any level ranging from system to component level.

5.1.2 Approaches to Modeling

The approach to modeling depends on the kind of information available and the goal of the modeling. There are two basic approaches to modeling failures, as indicated below.

1. Black-box approach: here the modeling is based solely on failure and censored data for similar items. This approach is used when there is very little understanding of the different mechanisms that lead to product failure or when the unit is too complex. This approach is also known as *data-based or empirical modeling*.
2. White-box approach: here the failure modeling at the component level is based on the different mechanisms that lead to failure. At the system level, the failure is done in terms of the failures of the different components. This approach is also known as *physics-based modeling*.

It should be noted that most statistical modeling incorporates features from both approaches and that the collaborative nature of such modeling is critical. The engineer or scientist needs to bring their expertise to develop the appropriate models in the white-box approach, whereas the statistician needs to be able to determine the appropriate data analysis for the given data set in its context.

5.1.3 First and Subsequent Failures

One needs to differentiate between the first failure and subsequent failures. The subsequent failures depend on the type of actions used to rectify the failures. In the case of a nonrepairable item (component, system or something in between), the failed item needs to be replaced by either a new or used item to make the product functional. In the case of a repairable item, the product can be made operational through the repair of the failed item. Three types of repair are indicated below:

1. Minimal repair, which restores the item to the condition just before failure;
2. Perfect repair (which makes the item as good as new); and
3. Imperfect repair that results in the item being better than it was prior to failure but not as good as new.

5.2 Black-Box Modeling Process

In the black-box approach to modeling failures, the data are the starting point that forms the basis for the model building. In this section we briefly discuss the different data types and then outline the modeling process.

5.2.1 Data Types

The data can be either item failure times or counts of item failures over an interval. In the former case, the data are continuous-valued and in the latter case they are integer-valued.

Lifetime data can be complete, censored or truncated. In the case of complete data, the data relate to the age at failure. With censored data, the lifetimes are only known to exceed some value(s). This could result from the item not having failed during the period of observation and hence still being operational for a certain length of time afterwards (right-censored data). When the data are the failures of an item over different disjoint time intervals we have grouped data. When failures of different components are pooled together, we have pooled data. In both cases, the data can be considered as categorical or, if they are in the form of counts, they are discrete-valued.

5.2.2 Modeling Process

The modeling process involves the following four steps.

Step 1: Exploratory analysis of data
Step 2: Model selection
Step 3: Parameter estimation
Step 4: Model validation

These are discussed further in Sects. 5.3–5.5.

5.3 One-Dimensional Black-Box Failure Modeling

In this section we give a brief review of one-dimensional failure modeling. The item under consideration can be the product or some component of the product.

5.3.1 Modeling First Failure

Let T denote the time to first failure. It is modeled by a failure distribution function, $F(t\,;\theta)$, which characterizes the probability $P\{T \le t\}$ and is defined as

$$F(t\,;\theta) = P(T \le t)\,,\ t \ge 0 \tag{5.1}$$

θ denotes the parameters (or parameter set) of the distribution function. If $F(t\,;\theta)$ is a differentiable function, then the failure density function, $f(t\,;\theta)$, is given by

$$f(t) = \mathrm{d}F(t;\theta)/\mathrm{d}t\,. \tag{5.2}$$

The survivor function, $\bar{F}(t)$, is given by

$$\bar{F}(t\,;\theta) = 1 - F(t\,;\theta) = P(T > t)\,. \tag{5.3}$$

The hazard function, $h(t\,;\theta)$, is given by

$$h(t\,;\theta) = f(t\,;\theta)/\bar{F}(t\,;\theta)\,. \tag{5.4}$$

The cumulative hazard function, $H(t\,;\theta)$, is given by

$$H(t\,;\theta) = \int_0^t h(u\,;\theta)\mathrm{d}u = -\log[1 - F(t\,;\theta)]\,. \tag{5.5}$$

Many different distributions have been used for modeling lifetimes. The shapes of the density and hazard functions depend on the form of the distribution and the parameter values.

Note: in the future we will omit θ for notational ease so that we have $h(t)$ instead of $h(t\,;\theta)$ and similarly for the other functions.

A commonly used model is the two-parameter Weibull distribution, which is given by

$$F(t\,;\theta) = 1 - \exp[-(t/\alpha)]^{\beta}\,,\ t \ge 0 \tag{5.6}$$

with $\theta = \{\alpha, \beta\}$. Here, α is the scale parameter and β is the shape parameter.

The Weibull models are a family of distributions derived from the two-parameter Weibull distribution. *Lai* et al. [5.5] discuss a few of these models and, for more details, see *Murthy* et al. [5.6]. Many other distributions have been used in modeling time to failure and these can be found in most books on reliability. See, for example, *Blischke* and *Murthy* [5.2], *Meeker* and *Escobar* [5.4], *Lawless* [5.3], *Nelson* [5.7] and *Kalbfleisch* and *Prentice* [5.8]. *Johnson* and *Kotz* [5.9, 10] give more details of other distributions that can be used for failure modeling.

5.3.2 Modeling Subsequent Failures

Minimal Repair

In minimal repair, the hazard function after repair is the same as that just before failure. In general, the repair time is small relative to the mean time between failures so that it can be ignored and the repairs treated as instantaneous. In this case, failures over time occur according to a nonhomogeneous Poisson point process with intensity function $\lambda(t) = h(t)$, the hazard function. Let $N(t)$

denote the number of failures over the interval $[0, t)$. Define

$$\Lambda(t) = \int_o^t \lambda(u)\,\mathrm{d}u \tag{5.7}$$

Then we have the following results:

$$P(N(t) = n) = \frac{\mathrm{e}^{-\Lambda(t)}[\Lambda(t)]^n}{n!}\,, n = 0\,,1\,,2\,,\ldots \tag{5.8}$$

and

$$E[N(t)] = \Lambda(t) \tag{5.9}$$

For more details, see *Nakagawa* and *Kowada* [5.11] and *Murthy* [5.12].

Perfect Repair

This is identical to replacement by a new item. If the failures are independent, then the times between failures are independent, identically distributed random variables from $F(t)$ and the number of failures over $[0, t)$ is a renewal process with

$$P\{N(t) = n\} = F^{(n)}(t) - F^{(n+1)}(t)\,, \tag{5.10}$$

where $F^{(n)}(t)$ is the n-fold convolution of $F(t)$ with itself, and

$$E[N(t)] = M(t)\,, \tag{5.11}$$

where $M(t)$ is the renewal function associated with $F(x\,;\theta)$ and is given by

$$M(t) = F(t) + \int_0^t M(t-u)\,\mathrm{d}F(u)\,. \tag{5.12}$$

For more on renewal processes, see *Cox* [5.13], *Cox* and *Isham* [5.14]) and *Ross* [5.15].

Imperfect Repair

Many different imperfect repair models have been proposed. See *Pham* and *Wang* [5.16] for a review of these models. In these models, the intensity function $\lambda(t)$ is a function of $\Im_t$, the history of failures over $[0, t)$. Two models that have received considerable attention are: (i) reduction in failure intensity, and (ii) virtual age. See *Doyen* and *Gaudoin* [5.17] for more on these two models.

5.3.3 Exploratory Data Analysis

The first step in constructing a model is to explore the data through plots of the data. By so doing, information can be extracted to assist in model selection. The plots can be either nonparametric or parametric and the plotting is different for perfect repair and imperfect repair situations. The data comprises both the failure times and the censored times.

Perfect Repair

Plot of Hazard Function (Nonparametric). The procedure (for complete or censored data) is as follows:

Divide the time axis into cells with cell i defined by $[t_i, t_{i+1})$, $i \geq 0$, $t_0 = 0$ and $t_i = i\delta$, where δ is the cell width. Define the following quantities:

N_i^{f} : Number of items with failure times in cell $i\,, i \geq 0$;

N_i^{c} : Number of items with censoring times in cell $i\,, i \geq 0$;

$N_i^{\mathrm{f}|ri}$: Number of failures in cells i and beyond $\left(= \sum_{j=i}^{\infty} N_j^{\mathrm{f}}\right)$.

Similarly define $N_i^{\mathrm{c}|ri}$ for censored data.

The estimator of the hazard function is given by

$$\hat{h}_i = \frac{N_i^{\mathrm{f}}}{N_i^{\mathrm{f}|ri} + N_i^{\mathrm{c}|ri}}\,, i \geq 0 \tag{5.13}$$

Plot of Density Function (Nonparametric). The simplest form of nonparametric density estimator is the histogram. Assuming the data is complete, the procedure is to calculate the relative frequencies for each cell,

$$\hat{f}_i = \frac{N_i^{\mathrm{f}}}{\sum_{j=0}^{\infty} N_j^{\mathrm{f}}}\,, \tag{5.14}$$

and then plot these against the cell midpoints. As histograms can be very unreliable for exploring the shape of the data, especially if the data set is not large, it is desirable to use more sophisticated density-function estimators (*Silverman* [5.18]).

Weibull Probability Plots (Parametric). The Weibull probability plot (WPP) provides a systematic procedure to determine whether one of the Weibull-based models is suitable for modeling a given data set or not, and is more reliable than considering just a simple histogram. It is based on the Weibull transformations

$$y = \ln\{-\ln[1 - F(t)]\} \quad \text{and} \quad x = \ln(t)\,. \tag{5.15}$$

The plot of y versus x gives a straight line if $F(t)$ is a two-parameter Weibull distribution.

Thus, if $F(t)$ is estimated for (complete) data from a Weibull distribution and the equivalent transformations and plot obtained, then a *rough* linear relationship should be evident. To estimate $F(t)$, we need an empirical estimate of $F(t_i)$ for each failure time t_i.

Assuming the t_i's are ordered, so that $t_1 \le t_2 \le \ldots \le t_n$, a simple choice (in the case of complete data) is to take the empirical distribution function

$$\hat{F}(t_i) = i/(n+1)\,. \tag{5.16}$$

We then plot $\hat{y}_i = \ln\left\{-\ln\left[1-\hat{F}(t_i)\right]\right\}$ versus $x_i = \ln(t_i)$ and assess visually whether a straight line could describe the points.

We illustrate by considering real data. The data refers to failure times and usage (defined through distance traveled between failures) for a component of an automobile engine over the warranty period given by three years and 60 000 miles. Here we only look at the failure times in the data set. Figure 5.1 shows a Weibull probability plot of the inter-failure times of a component that we shall call component C-1. This clearly shows a curved relationship and so a simple Weibull model would not be appropriate.

Note: the plotting of the data depends on the type of data. So, for example, the presence of censored observations would necessitate a change in the empirical failure estimates (see *Nelson* [5.7] for further details).

Minimal Repair

Plot of Cumulative Intensity Function (Nonparametric). The procedure is as follows: With δ and the cells defined as before, define the following:

M : Number of items at the start;
N_i^{f} : Total number of failures over $[0\,,\,i\delta)$;
M_i^{c} : Number of items censored in cell i;
λ_i : Cumulative intensity function till cell i.

The estimator of the cumulative intensity is given by

$$\hat{\lambda}_0 = \frac{N_0^{\mathrm{f}}}{M} \quad \text{and}$$
$$\hat{\lambda}_i = \frac{N_i^{\mathrm{f}} - \sum_{j=0}^{i-1} M_j^{\mathrm{c}}\hat{\lambda}_j}{\left[M - \sum_{j=0}^{i-1} M_j^{\mathrm{c}}\right]}\,,\ i \ge 1\,. \tag{5.17}$$

Graphical Plot (Parametric). When the failure distribution is a two-parameter Weibull distribution, from (5.9) we see that a plot of $y = \ln\{E[N(t)]/t\}$ versus $x = \ln(t)$ is a straight line. *Duane* [5.19] proposed plotting $y = \ln[N(t)/t]$ versus $x = \ln(t)$ to determine if a Weibull distribution is a suitable model or not to model a given data set. For a critical discussion of this approach, see *Rigdon* and *Basu* [5.20].

5.3.4 Model Selection

We saw in Fig. 5.1 that a simple Weibull model was clearly not adequate to model the failures of component C-1. However, there are many extensions of the Weibull model that can fit a variety of shapes. *Murthy* et al. [5.6] give a taxonomic guide to such models and give steps for model selection. This particular curve is suited to modeling with a mixture of two Weibull components. Figure 5.2 shows the WPP plot of Fig. 5.1 with the transformed probability curve for this mixture. (Details about estimating this curve are given in Sect. 5.3.5.) This seems to fit the pattern quite well, although it misses

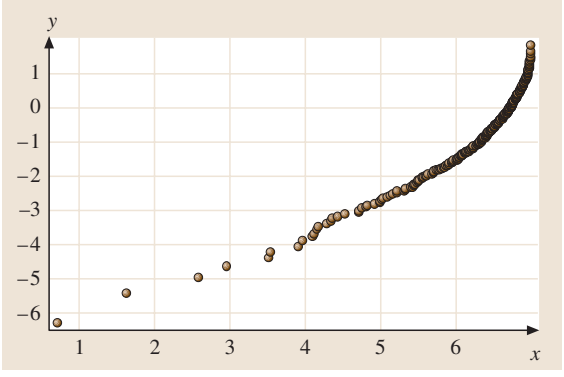


Fig. 5.1 WPP of days to failure of component C-1

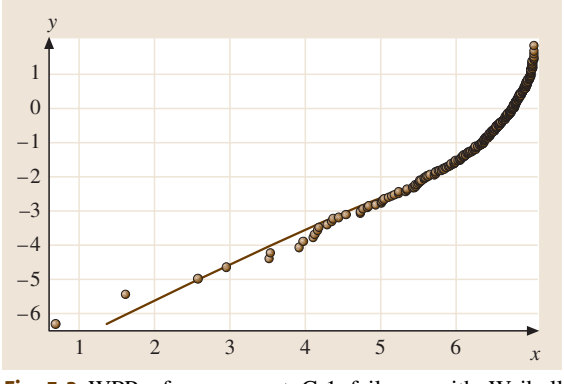


Fig. 5.2 WPP of component C-1 failures with Weibull mixture

the shape of the curve present in the few small failure times.

Figure 5.3 gives the empirical plot of the density function and the density function based on the mixture model. As can be seen, the model matches the data reasonably well. The plots illustrate the way in which the second Weibull component is being used. The nonparametric density estimate suggests that there is a small failure mode centered around 200 d. The second Weibull component, with a weight of 24.2%, captures these early failure times while the dominant component, with a weight of 75.8%, captures the bulk of the failures.

5.3.5 Parameter Estimation

The model parameters can be estimated either based on the graphical plots or by using statistical methods. Many different methods (method of moments, method of maximum likelihood, least squares, Bayesian and so on) have been proposed. The graphical methods yield crude estimates whereas the statistical methods are more refined and can be used to obtain confidence limits for the estimates. Most books on statistical reliability (some of which are mentioned in Sect. 5.3.1) deal with this topic in detail.

The parameters for the Weibull mixture model in Fig. 5.3 were estimated by minimizing the squared error between the points and the curve on the Weibull probability plot. The estimates are

$$\hat{p} = 0.242\,, \quad \hat{\beta}_1 = 1.07\,, \quad \hat{\beta}_2 = 4.32\,,$$
$$\hat{\eta}_1 = 381 \quad \text{and} \quad \hat{\eta}_2 = 839\,.$$

Similar estimates can be obtained without computer software using the graphical methods given by *Jiang* and *Murthy* [5.21].

Alternatively, we can use the standard statistical approach of maximum-likelihood estimation to get the parameter estimates. We find

$$\hat{p} = 0.303\,, \hat{\beta}_1 = 1.46\,, \hat{\beta}_2 = 5.38\,,$$
$$\hat{\eta}_1 = 383 \quad \text{and} \quad \hat{\eta}_2 = 870\,.$$

These values are less affected by the small failures times.

5.3.6 Model Validation

Validation of statistical models is highly dependent on the nature of the models being used. In many situations, it can simply involve an investigation of the shape of the data through plots such as quantile–quantile plots (for example, normal probability plots and WPP) and through tests for goodness of fit (general tests, such as the χ^2 goodness-of-fit test, or specific tests, such as the Anderson–Darling test of normality). Many introductory statistics texts cover these plots and tests (see, for example, *Vardeman* [5.22] and *D'Agostino* and *Stephen* [5.23]). In more complex situations, these approaches need to be used on residuals obtained after fitting a model involving explanatory variables. An alternative approach, which can be taken when the data set is large, is to take a random sample from the data set, fit the model(s) to this sub-sample and then evaluate (through plots and tests) how well the model fits the sub-sample consisting of the remaining data.

To exemplify model validation, 80% of the data was randomly taken and the mixed Weibull model above fitted. The fitted model was then compared using a WPP to the remaining 20% of the data. The upper top of Fig. 5.4 shows a Weibull plot of 80% of the failure data for component C-1, together with the Weibull mixture fit to the data. The remaining 20% of failure data are

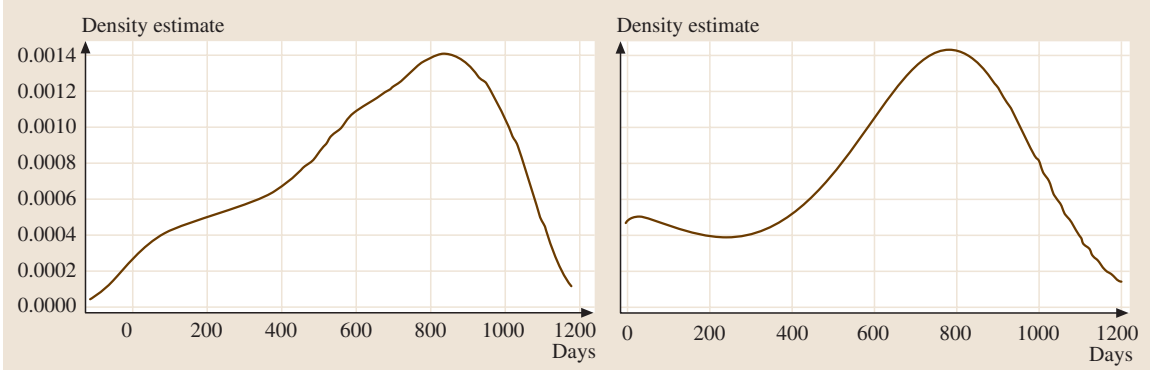


Fig. 5.3 Empirical density (*left*) and Weibull mixture density (*right*) for component C-1

plotted in the lower plot. The Weibull mixture curve with the same parameters as in the upper plot has been added here. Apart from the one short failure time, this curve seems to fit the test data quite well. This supports the use of the Weibull mixture for modeling the failures of this component.

5.4 Two-Dimensional Black-Box Failure Modeling

When failure depends on age and usage, one needs a two-dimensional failure model. Two different approaches (one-dimensional and two-dimensional) have been proposed and we discuss both of these in this section.

5.4.1 One-Dimensional Approach

Here, the two-dimensional problem is effectively reduced to a one-dimensional problem by treating usage as a random function of age.

Modeling First Failure

Let $X(t)$ denote the usage of the item at age t. In the one-dimensional approach, $X(t)$ is modeled as a linear function of t and so given by

$$X(t) = \Gamma t \tag{5.18}$$

where Γ, $0 \le \Gamma < \infty$, represents the usage rate and is a nonnegative random variable with a distribution function $G(r)$ and density function $g(r)$.

The hazard function, conditional on $\Gamma = r$ is given by $h(t|r)$. Various forms of $h(t|r)$ have been proposed; one such is the following polynomial function:

$$h(t|r) = \theta_0 + \theta_1 r + \theta_2 t + \theta_3 X(t) + \theta_4 t^2 + \theta_5 t X(t) \,. \tag{5.19}$$

The conditional distribution function for the time to first failure is given by

$$F(t|r) = 1 - \exp\left[-\int_0^t h(u|r)\,\mathrm{d}u \right] \,. \tag{5.20}$$

On removing the conditioning, we have the distribution function for the time to first failure, given by

$$F(t) = \int_0^\infty \left\{ 1 - \exp\left[-\int_0^t h(u|r)\,\mathrm{d}u \right] \right\} g(r)\,\mathrm{d}r \,. \tag{5.21}$$

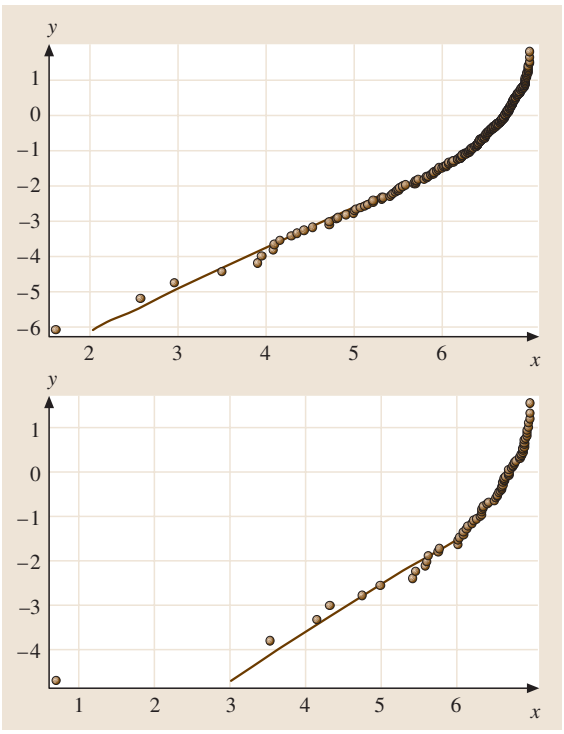


Fig. 5.4 Weibull plots of fitting data (*top*) and test data (*bottom*) for component C-1

Modeling Subsequent Failures

The modeling of subsequent failures, conditional on $\Gamma = r$, follows along lines similar to that in Sect. 5.3.2. As a result, under minimal repair, the failures over time occur according to a nonhomogeneous Poisson process with intensity function $\lambda(t|r) = h(t|r)$ and, under perfect repair, the failures occur according to the renewal process associated with $F(t|r)$.

The bulk of the literature deals with a linear relationship between usage and age. See, for example, *Blischke* and *Murthy* [5.1], *Lawless* et al. [5.24] and *Gertsbakh* and *Kordonsky* [5.25]. *Iskandar* and *Blischke* [5.26] deal with motorcycle data. See *Lawless* et al. [5.24] and *Yang* and *Zaghati* [5.27] for automobile warranty data analysis.

5.4.2 Two-Dimensional Approach

Modeling First Failure

Let T and X denote the system's age and usage at its first failure. In the two-dimensional approach to modeling, (T, X) is treated as a nonnegative bivariate random variable and is modeled by a bivariate distribution function

$$F(t\,,x) = P(T \le t\,, X \le x)\,; t \ge 0\,, x \ge 0\,. \quad (5.22)$$

The survivor function is given by

$$\bar{F}(t\,,x) = Pr(T > t\,, X > x) = \int_t^\infty \int_x^\infty f(u,v)\,\mathrm{d}v\,\mathrm{d}u\,. \quad (5.23)$$

If $F(t,x)$ is differentiable, then the bivariate failure density function is given by

$$f(t\,,x) = \frac{\partial^2 F(t\,,x)}{\partial t \partial x}\,. \quad (5.24)$$

The hazard function is defined as

$$h(t\,,x) = f(t\,,x)/\bar{F}(t\,,x)\,, \quad (5.25)$$

with $h(t\,,x)\delta t\delta x$ defining the probability that the first system failure will occur in the rectangle $[t\,, t+\delta t)\times[x\,, x+\delta x)$ given that $T > t$ and $X > x$. Note, however, that this is not the same as the probability that the first system failure will occur in the rectangle $[t\,, t+\delta t)\times[x\,, x+\delta x)$ given that it has not occurred *before* time t and usage x, which is given by $(f(t\,,x)/[1-F(t\,,x)])\,\delta t\delta x$.

Bivariate Weibull Models

A variety of bivariate Weibull models have been proposed in the literature. We indicate the forms of the models, and interested readers can obtain more details from *Murthy* et al. [5.6].

Model 1 [*Marshall* and *Olkin* [5.28]]

$$\bar{F}(t\,,x) = \exp\left\{-\left[\lambda_1 t^{\beta_1} + \lambda_2 x^{\beta_2} + \lambda_{12}\max(t^{\beta_1}\,, x^{\beta_2})\right]\right\}\,. \quad (5.26)$$

Model 2 [*Lee* [5.29]]

$$\bar{F}(t\,,x) = \exp\left\{-\left[\lambda_1 c_1^{\beta} t^{\beta} + \lambda_2 c_2^{\beta} x^{\beta} + \lambda_{12}\max\left(c_1^{\beta} t^{\beta}\,, c_2^{\beta} x^{\beta}\right)\right]\right\}\,. \quad (5.27)$$

Model 3 [*Lee* [5.29]]

$$\bar{F}(t\,,x) = \exp\left[-\lambda_1 t^{\beta_1} - \lambda_2 x^{\beta_2} - \lambda_0 \max(t\,,x)^{\beta_0}\right]\,. \quad (5.28)$$

Model 4 [*Lu* and *Bhattacharyya* [5.30]]

$$\bar{F}(t\,,x) = \exp\left\{-\left[(t/\theta_1)^{\beta_1/\delta} + (x/\theta_2)^{\beta_2/\delta}\right]^{\delta}\right\}\,, \quad (5.29)$$

$$\bar{F}(t\,,x) = \left[1 + \left(\left\{\exp\left[(t/\theta_1)^{\beta_1}\right] - 1\right\}^{1/\gamma} + \left\{\exp\left[(x/\theta_2)^{\beta_2}\right] - \right\}^{1/\gamma}\right)^{\gamma}\right]^{-1}\,, \quad (5.30)$$

$$\bar{F}(t\,,x) = \exp\left[-(t/\alpha_1)^{\beta_1} - (x/\alpha_2)^{\beta_2} - \delta h(t\,,x)\right]\,. \quad (5.31)$$

Different forms for the function of $h(t\,,x)$ yield a family of models. One form for $h(t\,,x)$ is the following:

$$h(t\,,x) = \left[(t/\alpha_1)^{\beta_1/m} + (x/\alpha_2)^{\beta_2/m}\right] \quad (5.32)$$

which results in

$$\bar{F}(t\,,x) = \exp\left\{-(t/\alpha_1)^{\beta_1} - (x/\alpha_2)^{\beta_2} - \delta\left[(t/\alpha_1)^{\beta_1/m} + (x/\alpha_2)^{\beta_2/m}\right]^m\right\}\,. \quad (5.33)$$

Two other variations are

$$\bar{F}(t\,,x) = \exp\left(-(t/\alpha_1)^{\beta_1} - (x/\alpha_2)^{\beta_2} - \delta\left\{1 - \exp\left[-(t/\alpha_1)^{\beta_1}\right]\right\} \times\left\{1 - \exp\left[-(x/\alpha_2)^{\beta_2}\right]\right\}\right)\,, \quad (5.34)$$

$$\bar{F}(t\,,x) = \left\{1 + \left[\left\{\exp\left[(t/\alpha_1)^{\beta_1}\right] - 1\right\}^{1/\gamma} + \left\{\exp\left[(x/\alpha_2)^{\beta_2}\right] - 1\right\}^{1/\gamma}\right]^{\gamma}\right\}^{-1}\,. \quad (5.35)$$

Model 5 (*Sarkar* [5.31])

$$\bar{F}(t\,,x) = \begin{cases} \exp\left(\left[-(\lambda_1+\lambda_{12})t^{\beta_1}\right] \times\left\{1 - \left[A\left(\lambda_2 t^{\beta 1}\right)\right]^{-\gamma} \times\left[A(\lambda_2 x^{\beta_2})\right]^{1+\gamma}\right\}\right)\,, \\ \qquad t \ge x > 0\,; \\ \exp\left(\left[-(\lambda_2+\lambda_{12})x^{\beta_2}\right] \times\left\{1 - \left[A(\lambda_1 x^{\beta_2})\right]^{-\gamma} \times\left[A(\lambda_1 t^{\beta_1})\right]^{1+\gamma}\right\}\right)\,, \\ \qquad x \ge t > 0\,; \end{cases} \quad (5.36)$$

where $\gamma = \lambda_{12}/(\lambda_1+\lambda_2)$ and $A(z) = 1-\mathrm{e}^{-z}$, $z>0$.
Model 6 [*Lee* [5.29]]

$$\bar{F}(t\,,x) = \exp\left[-\left(\lambda_1 t^{\beta_1}+\lambda_2 x^{\beta_2}\right)^{\gamma}\right]\,. \tag{5.37}$$

Comment: many other non-Weibull models can also be used for modeling. For more on this see *Johnson* and *Kotz* [5.32] and *Hutchinson* and *Lai* [5.33]. *Kim* and *Rao* [5.34], *Murthy* et al. [5.35], *Singpurwalla* and *Wilson* [5.36], and *Yang* and *Nachlas* [5.37] deal with two-dimensional warranty analysis.

Modeling Subsequent Failures

Minimal Repair. Let the system's age and usage at the j-th failure be given by t_j and x_j, respectively. Under minimal repair, we have that

$$h\left(t_j^+\,,x_j\right) = h\left(t_j^-\,,x_j\right)\,, \tag{5.38}$$

as the hazard function after repair is the same as that just before failure. Note that there is no change in the usage when the failed system is undergoing minimal repair.

Let $\{N(t\,,x): t\geq 0\,,x\geq 0\}$ denote the number of failures over the region $[0\,,t)\times[0\,,x)$. Unfortunately, as there is no complete ordering of points in two dimensions, there is no analogous result to that obtained for minimal repair in one dimension. In particular, the hazard rate does not provide an intensity rate at a point $(t\,,x)$ as the failure after the last failure prior to $(t\,,x)$ may be either prior to time t (though after usage x) or prior to usage x (though after time t), as well as possibly being after both time t and usage x. Hence, not only is it more difficult to obtain the distribution for $\{N(t,x): t\geq 0, x\geq 0\}$, it is also more difficult to obtain even the mean function for this process.

Perfect Repair. In this case, we have a two-dimensional renewal process for system failures and the following results are from *Hunter* [5.38]:

$$p_n(t\,,x) = F^{(n)}(t\,,x) - F^{(n+1)}(t\,,x)\,,\, n\geq 0\,, \tag{5.39}$$

where $F^{(n)}(t\,,x)$ is the n-fold bivariate convolution of $F(t\,,x)$with itself. The expected number of failures over $[0\,,t)\times[0\,,x)$ is then given by the solution of the two-dimensional integral equation

$$M(t\,,x) = F(t\,,x) + \int_0^t\int_0^x M(t-u\,,x-v) \times f(u\,,v)\,\mathrm{d}v\,\mathrm{d}u\,. \tag{5.40}$$

Imperfect Repair. This has not been studied and hence is a topic for future research.

Comparison with 1-D Failure Modeling

For the first failure, in the one-dimensional failure modeling, we have

$$F(t) + \bar{F}(t) = 1\,, \tag{5.41}$$

and

$$\bar{F}(t) = \exp\left[-\int_0^t h(u)\,\mathrm{d}u\right]\,. \tag{5.42}$$

In two-dimensional failure modeling, however, we have

$$F(t\,,x) + \bar{F}(t\,,x) < 1\,, \tag{5.43}$$

since

$$F(t\,,x)+\bar{F}(t\,,x)+P\,(T\leq t\,,X>x) +P\,(T>t\,,X\leq x) = 1\,. \tag{5.44}$$

A Numerical Example

We confine our attention to a model proposed by *Lu* and *Bhattacharyya* [5.30], where the survivor function is given by (5.31) with $h(t\,,x)$ given by (5.32) with $\alpha_1\,,\alpha_2\,,\beta_1\,,\beta_2>0, \delta\geq 0$ and $0<m\leq 1$. If $m=1$ then the hazard function is given by

$$h(t\,,x) = (1+\delta)^2\frac{\beta_1}{\alpha_1}\left(\frac{t}{\alpha_1}\right)^{\beta_1-1}\frac{\beta_2}{\alpha_2}\left(\frac{t}{\alpha_2}\right)^{\beta_2-1}\,. \tag{5.45}$$

Let the model parameters be as follows:

$$\alpha_1 = 2\,,\alpha_2 = 3\,,\beta_1 = 1.5\,,\beta_2 = 2.0\,,$$
$$\delta = 0.5\,,m = 1$$

The units for age and usage are years and 10 000 km, respectively. The expected age and usage at first system failure are given by

$E(T_1) = \theta_1\Gamma(1/\beta_1+1) = 1.81$ (years) and
$E(X_1) = \theta_2\Gamma(1/\beta_2+1) = 2.66$ (10^3 km).

Figure 5.5 is a plot of the survivor function $\bar{F}(t\,,x)$ and Fig. 5.6 is a plot of the hazard function $h(t\,,x)$. Note that $h(t\,,x)$ increases as t (age) and x (usage) increase, since β_1 and β_2 are greater than 1.

Replacement. The expected number of system failures in the rectangle $[0\,,t)\times[0\,,x)$ under replacement is given by the renewal function $M(t\,,x)$ in (5.40). Figure 5.7 is

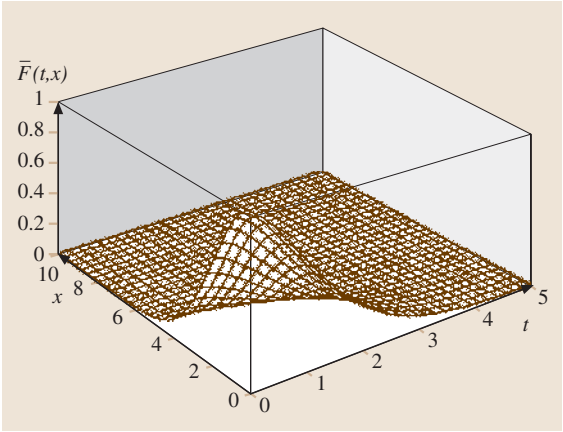


Fig. 5.5 Plot of the survivor function $\bar{F}(t\,,x)$

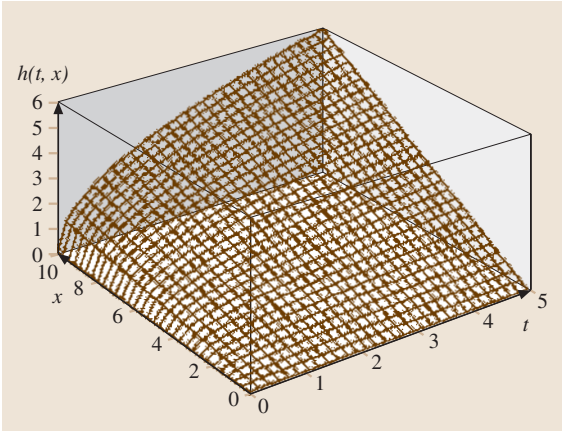


Fig. 5.6 Plot of the hazard function $h(t\,,x)$

a plot of $M(t\,,x)$, obtained using the two-dimensional renewal-equation solver from *Iskandar* [5.39].

5.4.3 Exploratory Data Analysis

In the one-dimensional case, the presence of censored observations causes difficulties in estimating the various functions (hazard rate, density function). When usage is taken into account, these difficulties are exacerbated, due to the information about usage being only observed at failure times. In particular, models which build conditional distributions for the failure times given usage (or usage rates) have to determine a strategy for assigning the censored failure times to some usage (or usage group).

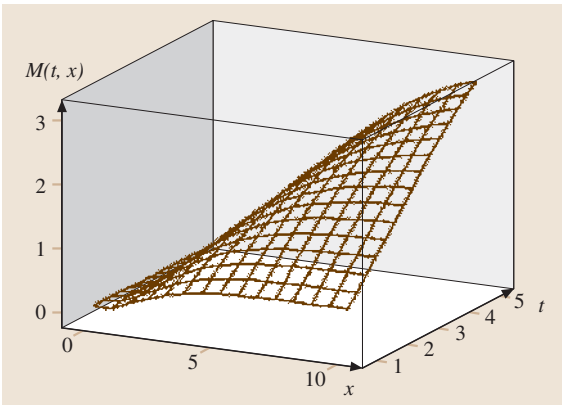


Fig. 5.7 Plot of the renewal function $M(t\,,x)$

1-D Approach

Perfect Repair. Firstly, we group the data into different groups based on the usage rate. Each group has a mean usage rate and the data is analyzed using the approach discussed in Sect. 5.3. This yields the model for the failure distribution *conditioned* on the usage rate. One then needs to determine whether the model structure is the same for different usages or not and whether the linear relationship [given by (5.16)] is valid or not. Next, exploratory plots of the usage rate need to be obtained to determine the kind of distribution appropriate to model the usage rate.

If the conditional failure distributions are two-parameter Weibull distributions then the WPP plots are straight lines. If the shape parameters do not vary with usage rate, then the straight lines are parallel to each other. One can view usage in a manner similar to stress level and use accelerated life-test models [*Nelson* [5.7]] to model the effect of usage on failure.

Imperfect Repair. The plotting (for a given usage rate) follows along the lines discussed in Sect. 5.3.3 and this allows one to determine the distribution appropriate to model the data. Once this is done, one again needs to examine exploratory plots of the usage rate to decide on the appropriate model.

2-D Approach

We confine our discussion to the case of perfect repair.

Plot of Hazard Function (Nonparametric Approach). We divide the region into rectangular cells. Cell $(i\,,j)$ is given by $[i\delta_1\,,(i+1)\delta_1)\times[j\delta_2\,,(j+1)\delta_2)$, where δ_1 and δ_2 are the cells' width and height respectively.

Let us define:

N_{ij}^{f} : Number of items with failures times in cell $(i\,,\,j)\,, i \geq 0\,,\, j \geq 0\,;$

N_{ij}^{c} : Number of items with censoring times in cell $(i\,,\,j)\,, i \geq 0\,,\, j \geq 0\,;$

$N_{ij}^{\mathrm{f|sw}}$: Number of failures in cells to the southwest of

$$\mathrm{cell}(i\,,\,j)\left(=\sum_{i'=0}^{i-1}\sum_{j'=0}^{j-1} N_{i'j'}^{\mathrm{f}}\right)\,;$$

$N_{ij}^{\mathrm{f|ne}}$: Number of failures in cells to the northeast of

$$\mathrm{cell}(i-1\,,\,j-1)\left(=\sum_{i'=i}^{\infty}\sum_{j'=j}^{\infty} N_{i'j'}^{\mathrm{f}}\right)\,.$$

Similarly define $N_{ij}^{\mathrm{c|ne}}$ and $N_{ij}^{\mathrm{c|sw}}$ for censored data.

A nonparametric estimator of the hazard function is

$$\hat{h}_{ij} = \frac{N_{ij}^{\mathrm{f}}}{N_{ij}^{\mathrm{f|ne}} + N_{ij}^{\mathrm{c|ne}}}\,, i \geq 0\,,\, j \geq 0\,. \tag{5.46}$$

Plot of Renewal Function (Nonparametric Approach). A simple estimator of the renewal function in the case of complete data is given by the partial mean function over the cells; that is,

$$\hat{M}(t_i\,,\,x_i) = \frac{N_{ij}^{\mathrm{f|sw}}}{N}\,, \tag{5.47}$$

where N is the total number of observations. A contour plot of this versus t and x can then be obtained.

5.4.4 Model Selection

To determine if the estimate of the renewal function above corresponds to the renewal function for a given model, plots similar to quantile–quantile plots can be investigated. Firstly, the renewal function for the given model is estimated and then its values are plotted against the corresponding values of the nonparametric estimator above. If a (rough) linear relationship is present, then this would be indicative that the model is reasonable.

5.4.5 Parameter Estimation and Validation

Once an appropriate model for $h(t\,,\,x) = f(t\,,\,x)/\bar{F}(t\,,\,x)$ is determined, estimation of the parameters can be carried out using standard statistical procedures (least squares, maximum likelihood, and so on) in a similar fashion to the one-dimensional case, although we are unaware of any equivalent graphical methods which may be used. Similarly, model validation can be carried out as before. It should be noted that the procedure indicated in the previous section can also be used to validate the model, if not used to select it. In fact, a common approach when faced with a complex model may be to fit the model using an estimation procedure such as maximum-likelihood estimation (or generalized least squares using an empirical version of a functional such as the renewal function), then investigating the relationship between some other functional of the model and its empirical version. This area requires further investigation.

5.5 A New Approach to Two-Dimensional Modeling

One of the attractions of the one-dimensional approach taken in Sect. 5.4.1 is that it matches the manner in which the failures occur in practice; that is, the expectation is that the failure time is dependent on the amount of usage of the item—different usage leads to different distributions for the time until failure, with these distributions reflecting the ordering that higher usage leads to shorter time until failure. However, usage may vary over time for individual items and the one-dimensional approach does not allow for this aspect. The model described in this section overcomes this shortcoming by allowing usage to vary between failures.

5.5.1 Model Description

For convenience, consider a single item. Suppose that T_i is the time until the i-th failure and that X_i is the total usage at the i-th failure. Analogously to the one-dimensional approach, let $\Gamma_i = (X_i - X_{i-1})/(T_i - T_{i-1})$ be the usage rate between the $(i-1)$th and the i-th failures. Assuming that these usage rates are independent and come from a common distribution (an oversimplification but a useful starting point for developing models), the marginal distribution of the usage rates can be modeled, followed by the times until failure modeled for different usage rates after each failure in

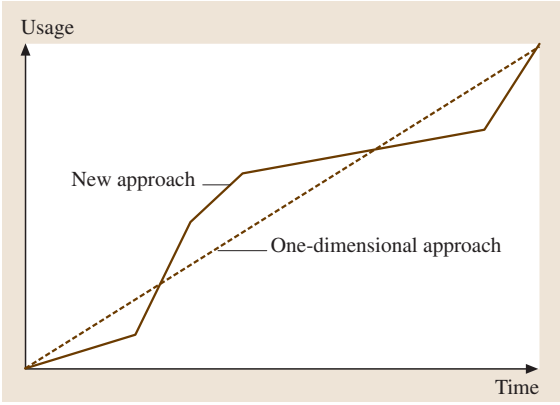


Fig. 5.8 Plot of usage versus time

a similar manner to the one-dimensional approach. This approach combines the two approaches discussed earlier. Figure 5.8 shows the plots of usage versus time for the proposed model and the model based on the one-dimensional approach.

Note that a more general approach is to model the usage as a cumulative stochastic process. This approach has received some attention. See *Lawless* et al. [5.24] and *Finkelstein* [5.40] and the references cited therein for further details.

5.5.2 An Application

We illustrate by considering the failure and usage data for component C-1 over the warranty period.

Modeling the Usage Rates

Before investigating the relationship between usage rate and time to failure, it is worthwhile investigating the days to failure and usage at failure for claims made within the warranty period. This is shown in Fig. 5.9. Only three of the failures were a second failure on the component; all of the others are the first failure since manufacture. There are three considerations to take into account when interpreting Fig. 5.9.

Firstly, the censoring by both time and usage ensures that only the initial part of the bivariate distribution of usage at failure and time to failure can be explored and the relationship between usage at failure and time to failure is distorted. Secondly, the proportions of components according to usage rate vary considerably, with very few components having high or low usage rates (as would be expected). Lastly, there are a greater number of short failure times than might be expected, suggesting that many early failures may not be related to usage and are more likely to be the result of quality-related problems during production. What seems like a linear trend (around the line $Usage = 50\times Days$) is not valid in light of the above discussion.

Figure 5.10 looks at the conditional distribution of days to failures against the usage rate (miles/day) averaged over the time before the claim. Again, care needs to be taken in interpreting this plot. In the left of this plot, censoring due to time and the low number of components having low usage rate has distorted the distribution of failure times for each usage rate. From a usage rate of around 60 km/d, the key feature of the plot is the censoring due to reaching the usage limit. Thus, although it would be expected that the failure-time distribution would be concentrated around a decreasing

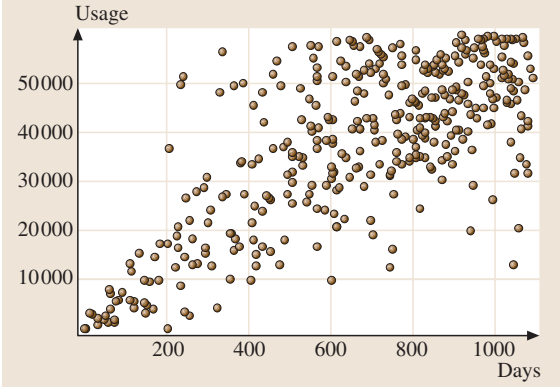


Fig. 5.9 Plot of days to failure and usage at failure for claims within warranty for component C-1

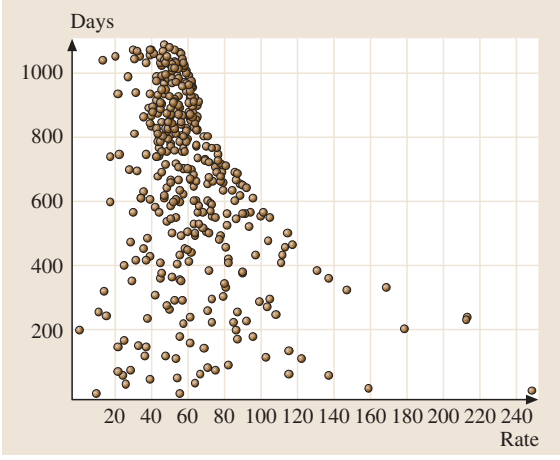


Fig. 5.10 Plot of days to failure against usage rate for component C-1

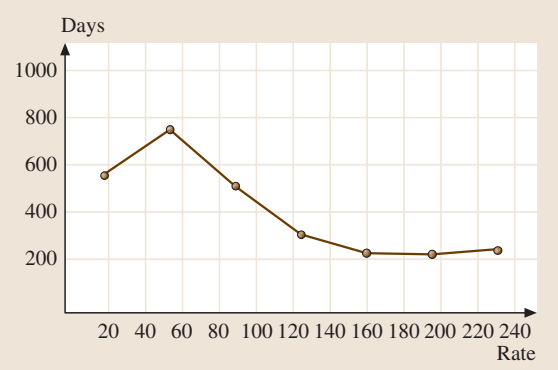


Fig. 5.11 Mean time to failure for usage-rate bands for component C-1

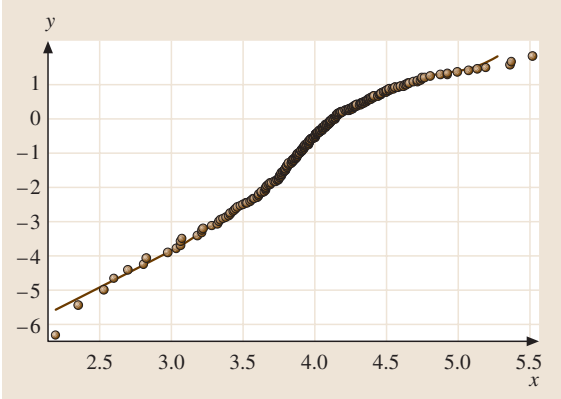


Fig. 5.12 WPP of usage rates for component C-1 with Weibull mixture

mean as usage rate increases, this is exaggerated by the censoring.

Figure 5.11 shows a plot of the mean time to failure for a split of failures into usage-rate bands. In this plot, we see the effects of the censoring, as indicated above. Thus, the mean time to failure actually increases in the low-usage-rate regime. For usage rate greater than 60 km/d, the mean time to failure decreases as the usage rate increases, as is expected (although, as discussed above, this is exaggerated by the censoring due to reaching the usage limit).

Figure 5.12 is a WPP plot of the usage rate. The plot indicates that a Weibull mixture involving two subpopulations is appropriate to model the usage rate. The parameter estimates of the fitted curve in Fig. 5.12 are

$$\hat{p} = 0.647\,,\ \hat{\beta}_1 = 5.50\,,\ \hat{\beta}_2 = 1.99\,,$$
$$\hat{\eta}_1 = 57.7\,,\ \hat{\eta}_2 = 75.7\,.$$

These give mean usage rates to failure of 53.3 km/d and 75.7 km/d, respectively.

Figure 5.13 gives the empirical plot of the density function and the density function based on the mixture model. As can be seen, the model matches the data reasonably well. The model estimates that around 65% of the failures come from a subpopulation with a mean usage rate of 53.3 km/d, giving a dominant peak in the observed density. The other subpopulation, with a higher mean usage rate of 75.7 km/d, accounts for the extra failures occurring for usage rates between around 50 and 100 km/d.

For different usage rates (one for each band in Fig. 5.11) one can obtain the conditional failure distribution $F(t|r)$ in a manner similar to that in Sect. 5.3. It is important to note that this ignores the censored data and as such would yield a model that gives conservative estimates for the conditional mean time to failure. Combining this with the distribution function for the usage rate yields the two-dimensional failure model that can be used to find solutions to decision problems.

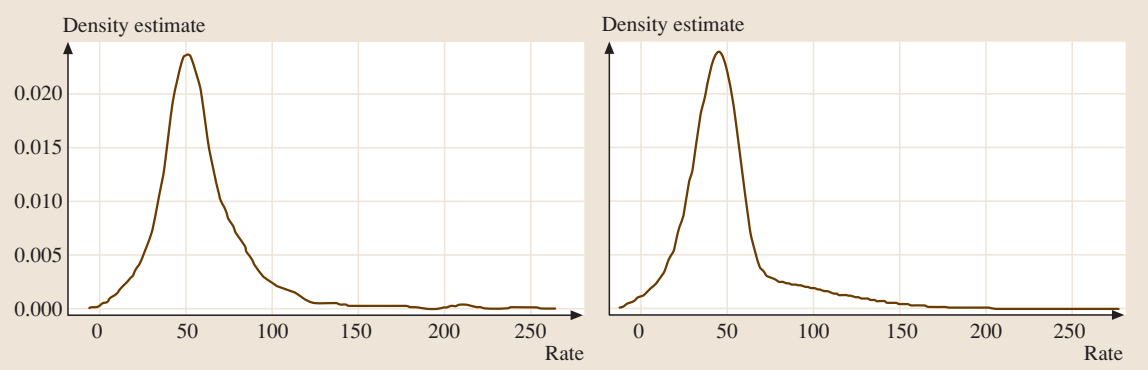


Fig. 5.13 Empirical density (*left*) and Weibull mixture density (*right*) for component C-1 usage rates

5.6 Conclusions

In this chapter we have looked at two-dimensional failure modeling. We have discussed the two approaches that have been proposed and suggested a new approach. However, there are several issues that need further study. We list these and hope that it will trigger more research in the future.

1. Different empirical plotting of two-dimensional failure data.
2. Study of models based on the two-dimensional approach and how this can be used in conjunction with the empirical plots to help in model selection.
3. Further study of the models based on the new approach discussed in Sect. 5.6.
4. Most failure data available for modeling is the data collected for products sold with two-dimensional warranties. In this case, the warranty can cease well before the time limit due to the usage limit being exceeded. This implies censored data with uncertainty in the censoring. This topic has received very little attention and raises several challenging statistical problems.

References

5.1 W. R. Blischke, D. N. P. Murthy: *Warranty Cost Analysis* (Marcel Dekker, New York 1994)
5.2 W. R. Blischke, D. N. P. Murthy: *Reliability* (Wiley, New York 2000)
5.3 J. F. Lawless: *Statistical Models and Methods for Lifetime Data* (Wiley, New York 1982)
5.4 W. Q. Meeker, L. A. Escobar: *Statistical Methods for Reliability Data* (Wiley, New York 1998)
5.5 C. D. Lai, D. N. P. Murthy, M. Xie: Weibull Distributions and Their Applications. In: *Springer Handbook of Engineering Statistics*, ed. by Pham (Springer, Berlin 2004)
5.6 D. N. P. Murthy, M. Xie, R. Jiang: *Weibull Models* (Wiley, New York 2003)
5.7 W. Nelson: *Applied Life Data Analysis* (Wiley, New York 1982)
5.8 J. D. Kalbfleisch, R. L. Prentice: *The Statistical Analysis of Failure Time Data* (Wiley, New York 1980)
5.9 N. L. Johnson, S. Kotz: *Distributions in Statistics: Continuous Univariate Distributions I* (Wiley, New York 1970)
5.10 N. L. Johnson, S. Kotz: *Distributions in Statistics: Continuous Univariate Distributions II* (Wiley, New York 1970)
5.11 T. Nakagawa, M. Kowada: Analysis of a system with minimal repair and its application to a replacement policy, Eur. J. Oper. Res. **12**, 176–182 (1983)
5.12 D. N. P. Murthy: A note on minimal repair, IEEE Trans. Reliab. **40**, 245–246 (1991)
5.13 D. R. Cox: *Renewal Theory* (Methuen, London 1967)
5.14 D. R. Cox, V. Isham: *Point Processes* (Chapman-Hall, New York 1980)
5.15 S. M. Ross: *Stochastic Processes* (Wiley, New York 1983)
5.16 H. Pham, H. Wang: Imperfect Maintenance, Eur. J. Oper. Res. **94**, 438–452 (1996)
5.17 L. Doyen, O. Gaudoin: Classes of imperfect repair models based on reduction of failure intensity function or virtual age, Reliab. Eng. Syst. Saf. **84**, 45–56 (2004)
5.18 B. W. Silverman: *Density Estimation for Statistics and Data Analysis* (Chapman Hall, London 1986)
5.19 J. T. Duane: Learning curve approach to reliability monitoring, IEEE Trans. Aerosp. **40**, 563–566 (1964)
5.20 S. E. Rigdon, A. P. Basu: *Statistical Methods for the Reliability of Repairable Systems* (Wiley, New York 2000)
5.21 R. Jiang, D. N. P. Murthy: Modeling failure data by mixture of two Weibull distributions, IEEE Trans. Reliab. **44**, 478–488 (1995)
5.22 S. B. Vardeman: *Statistics for Engineering Problem Solving* (PWS, Boston 1993)
5.23 R. B. D'Agostino, M. A. Stephens: *Goodness-of-Fit Techniques* (Marcel Dekker, New York 1986)
5.24 J. F. Lawless, J. Hu, J. Cao: Methods for the estimation of failure distributions and rates from automobile warranty data, Lifetime Data Anal. **1**, 227–240 (1995)
5.25 I. B. Gertsbakh, H. B. Kordonsky: Parallel time scales and two-dimensional manufacturer and individual customer warranties, IEE Trans. **30**, 1181–1189 (1998)
5.26 B. P. Iskandar, W. R. Blischke: Reliability and Warranty Analysis of a Motorcycle Based on Claims Data. In: *Case Studies in Reliability and Maintenance*, ed. by W. R. Blischke, D. N. P. Murthy (Wiley, New York 2003) pp. 623–656
5.27 G. Yang, Z. Zaghati: *Two-dimensional reliability modelling from warranty data*, Ann. Reliab. Maintainab. Symp. Proc. 272–278 (IEEE, New York 2002)
5.28 A. W. Marshall, I. Olkin: A multivariate exponential distribution, J. Am. Stat. Assoc. **62**, 30–44 (1967)

5.29 L. Lee: Multivariate distributions having Weibull properties, J. Multivariate Anal. **9**, 267–277 (1979)

5.30 J. C. Lu, G. K. Bhattacharyya: Some new constructions of bivariate Weibull Models, Ann. Inst. Stat. Math. **42**, 543–559 (1990)

5.31 S. K. Sarkar: A continuous bivariate exponential distribution, J. Am. Stat. Assoc. **82**, 667–675 (1987)

5.32 N. L. Johnson, S. Kotz: *Distributions in Statistics: Continuous Multivairate Distributions* (Wiley, New York 1972)

5.33 T. P. Hutchinson, C. D. Lai: *Continuous Bivariate Distributions, Emphasising Applications* (Rumsby Scientific, Adelaide 1990)

5.34 H. G. Kim, B. M. Rao: Expected warranty cost of a two-attribute free-replacement warranties based on a bi-variate exponential distribution, Comput. Ind. Eng. **38**, 425–434 (2000)

5.35 D. N. P. Murthy, B. P. Iskandar, R. J. Wilson: Two-dimensional failure free warranties: Two-dimensional point process models, Oper. Res. **43**, 356–366 (1995)

5.36 N. D. Singpurwalla, S. P. Wilson: Failure models indexed by two scales, Adv. Appl. Probab. **30**, 1058–1072 (1998)

5.37 S.-C. Yang, J. A. Nachlas: Bivariate reliability and availability modeling, IEEE Trans. Reliab. **50**, 26–35 (2001)

5.38 J. J. Hunter: Renewal theory in two dimensions: Basic results, Adv. Appl. Probab. **6**, 376–391 (1974)

5.39 B. P. Iskandar: *Two-Dimensional Renewal Function Solver*, Res. Rep. No. 4/91 (Dept. of Mechanical Engineering, Univ. Queensland, Queensland 1991)

5.40 M. S. Finkelstein: Alternative time scales for systems with random usage, IEEE Trans. Reliab. **50**, 261–264 (2004)

6. Prediction Intervals for Reliability Growth Models with Small Sample Sizes

The first section of this chapter provides an introduction to the types of test considered for this growth model and a description of the two main forms of uncertainty encountered within statistical modelling, namely aleatory and epistemic. These two forms are combined to generate prediction intervals for use in reliability growth analysis.

The second section of this chapter provides a historical account of the modelling form used to support prediction intervals. An industry-standard model is described and will be extended to account for both forms of uncertainty in supporting predictions of the time to the detection of the next fault.

The third section of this chapter describes the derivation of the prediction intervals. The approach to modelling growth uses a hybrid of the Bayesian and frequentist approaches to statistical inference. A prior distribution is used to describe the number of potential faults believed to exist within a system design, while reliability growth test data is used to estimate the rate at which these faults are detected.

After deriving the prediction intervals, the fourth section of this chapter provides an analysis of the statistical properties of the underlying distribution for a range of small sample sizes.

The fifth section gives an illustrative example used to demonstrate the computation and interpretation of the prediction intervals within a typical product development process.

The strengths and weaknesses of the process are discussed in the final section.

Predicting the time until a fault will be detected in a reliability growth test is complex due to the interaction of two sources of uncertainty, and hence often results in wide prediction intervals that are not practically credible. The first source of variation corresponds to the selection of an appropriate stochastic model to explain the random nature of the fault detection process. The second source of uncertainty is associated with the model itself, as even if the underlying stochastic process is known, the realisations will be random.

Statistically, the first source of uncertainty can be measured through confidence intervals. However, using these confidence intervals for prediction can result in naive underestimates of the time to realise the next failure because they will only infer the mean time to the next fault detection rather than the actual time of the fault detection.

Since the confidence in parameter estimates increases as sample size increase, the degree of underestimation arising from the use of confidence rather than prediction intervals will be lower for large sample sizes compared with small sample sizes. Yet, in reliability growth tests it is common, indeed desirable, to have a small number of failures. Therefore there is a need for prediction intervals, although their construction is more challenging for small samples since they are driven by the second source of variation.

The type of reliability growth test considered will be of the form test, analyse and fix (TAAF). This implies that a system is tested until it fails, at which time analy-

sis is conducted to investigate potential causes and hence identify the source of the fault. Once found, a corrective action is implemented and the 'new' system design is returned to test. This cyclical process is repeated until all weaknesses have been flushed out and the system design is deemed mature. The data generated through testing are analysed using reliability growth models and the information generated is used to support product development decisions. Examples of these data are the duration of the growth test to achieve the required reliability or the efficacy of the stresses experienced by the prototype designs during test within a specified test plan.

Procedures for the construction of prediction intervals for the time to realise the next failure are developed for a standard reliability growth model called the modified IBM model. This model is particularly suited for test situations where few data exist and some expert engineering judgement is available. The model consists of two parameters: one that reflects the number of faults within the system design; and a second that reflects the rate at which the faults are detected on test. Estimation procedures have been developed as a mixture of Bayesian and frequentist methods. Processes exist to elicit prior distributions describing the number of potential faults within a design. However prior distributions describing the test time by which these faults will be realised are more challenging to elicit and, as such, inference for this parameter is supported by data observed from the test.

Section 6.1 provides background history and a description of the model and its underlying assumptions. In Sect. 6.2, prediction intervals are derived for this model from the frequentist perspective. Section 6.3 presents an analysis of the statistical properties of the proposed estimators, while Sect. 6.4 provides an illustrative example of their application to a typical reliability growth test. Finally the use of such procedures is discussed in Sect. 6.5.

6.1 Modified IBM Model – A Brief History

The IBM model was proposed by [6.1] and was the first reliability growth model to represent formally two different types of failures: namely those that result in a corrective action to the system; and those that result in a replacement of a component part. By formally accounting for failures that occur but which are not addressed at the system level, the failure rate is estimated by an asymptote corresponding to a residual failure rate relating to faults that would have been detected and corrected, but were not, due to the termination of testing. The model was developed assuming the following differential equation

$$\frac{\mathrm{d}D(t)}{\mathrm{d}t} = -\mu D(t)\,, \quad \mu, t > 0\,, \tag{6.1}$$

where $D(t)$ represents the number of faults remaining in the system at accumulated test time t. Therefore the expected number of faults detected by accumulated test time t is

$$N(t) = D(0)\left(1 - \mathrm{e}^{-\mu t}\right)\,, \quad \mu, t > 0\,. \tag{6.2}$$

The model proposes that spurious failures would be realised according to a homogeneous Poisson process independent of the fault detection process. This latter process is not of direct concern to the model developed here and hence we do not develop it further. Instead we focus on the fault detection process only.

The deterministic approach to reliability growth modelling was popular in the late 1950s and early 1960s. However, this approach was superseded by the further development and popularity of statistical methods based upon stochastic processes. This is because the deterministic arguments relating to reliability growth could just as easily be interpreted through probabilistic methods, which also facilitated more descriptive measures for use in test programmes.

Cozzolino [6.2] arrived at the same form for an intensity function, $\iota(t)$, assuming that the number of initial defects within a system, $D(0)$, has a Poisson distribution with mean λ and that the time to failure of each initial defect followed an exponential distribution with hazard rate μ.

$$\iota(t) = \lambda\mu\,\mathrm{e}^{-\mu t}\,, \quad \mu\,, \lambda\,, t > 0 \tag{6.3}$$

implying that

$$E\left[N(t)\right] = \lambda\left(1 - \mathrm{e}^{-\mu t}\right)\,. \tag{6.4}$$

Jelenski and *Moranda* [6.3] proposed a model to describe the detection of faults within software testing assuming that there were a finite number of faults [i.e. $D(0)$] within the system and that the time between the detection of the i-th and $(i-1)$-th (i.e. W_i) fault had the following exponential cumulative distribution function

(CDF):

$$F(w_i) = 1 - \mathrm{e}^{-[D(0)-i+1]\,\mu\, w_i}\,,$$
$$i = 1, 2, \ldots, D(0)\,; \quad w_i, \mu > 0\,. \tag{6.5}$$

This model is the most referenced software reliability model [6.4]. It is assumed for this model that there exist $D(0)$ faults within the system, whose failure times are all independently and identically exponentially distributed with a hazard rate μ. It is interesting to note the similarities between this model, Cozzolino's model (6.4) and the IBM (6.2) deterministic model; the mean number of faults detected at accumulated test time t is the same in all three models.

Jelenski and Moranda advocated that maximum likelihood estimators (MLEs) be pursued for this model. However it was shown by *Forman* and *Singpurwalla* [6.5], and later by *Littlewood* and *Verrall* [6.6], that the MLEs were inconsistent, often did not exist or produced confidence regions so large that they were practically meaningless.

Meinhold and *Singpurwalla* [6.7] proposed an adaptation with a Poisson prior distribution to estimate the number of faults that will ultimately be exposed. This is similar to *Cozzolino* [6.2] but from a Bayesian perspective, so the variability described through the prior distribution captures knowledge uncertainty in the number of faults within the system design only. This Bayesian approach results in the following estimating equation for μ:

$$\hat{\mu} = \frac{j}{\lambda t' \mathrm{e}^{-t'\mu} + \sum_{i=1}^{j} t_i}\,, \quad t_1 < t_2 < \ldots < t_j \le t'\,, \tag{6.6}$$

where: t_i is the time of the i-th fault detected on test, t' is the test time when the analysis is conducted, and j is the number of faults detected by time t'.

This Bayesian adaptation was further explored in [6.8] and its advantages over the industry-standard model, the so-called power-law model [6.9, 10], was demonstrated. A process was developed to elicit the prior distribution for this model in [6.11] and procedures for estimating confidence intervals for μ were developed in [6.12]. Extensions of this model to assess the cost effectiveness of reliability growth testing are addressed in [6.13], and for managing design processes see [6.14]. This model is incorporated into the international standard, IEC 61164, as the modified IBM model.

The model assumes that there are an unknown but fixed number of faults within a system design. It is further assumed that the system undergoes a TAAF regime that gives rise to times to detect faults that are independently and identically exponentially distributed. A prior distribution describing the number of faults existing within the system design is assumed to be Poisson. Finally, it is assumed that when a fault is identified it is removed completely and no new fault is introduced into the design.

6.2 Derivation of Prediction Intervals for the Time to Detection of Next Fault

We assume that a system contains $D(0)$ faults. The time to detect a fault is exponentially distributed with hazard rate μ, and a prior distribution exists on $D(0)$, in the form of a Poisson distribution with mean λ. It is further assumed that j faults will be observed on test at times $t_1, \ldots t_j$ and we seek a prediction interval for the time to detect the next fault. Let x denote the time between t_j and t_{j+1} and assume that the times to detection are statistically independent.

The statistic R is defined as the ratio of x to the sum of the first j fault detection times, denoted by T:

$$R = \frac{x}{T}\,. \tag{6.7}$$

First, the distribution of T is derived. T is the sum of the first j order statistics from a sample of $D(0)$ independent and identically exponentially distributed random variables with hazard rate μ. Thus, the time between any two consecutive such order statistics are exponentially distributed with hazard rate $[D(0) - i + 1]\,\mu$. Moreover, the times between successive order statistics are independent. For a derivation of this results see [6.15]. Therefore T can be expressed as a weighted sum of independent and identically distributed exponential random variables with hazard rate μ. We denote these random variables as W_i in the following:

$$T = \sum_{i=1}^{j} t_i = \sum_{i=1}^{j} \frac{j-i+1}{N-i+1} W_i\,. \tag{6.8}$$

As an exponential random variable is closed under scale transformation, we can consider T, as expressed in (6.8), as a sum of independent exponential random variables with different hazard rates. As such, using *Goods* formula [6.16] we can express the CDF of T, conditional on there being j faults realised and the design initially having $D(0)$ faults, as:

$$\Pr\left(\sum_{i=1}^{j}\frac{(j-i+1)\,W_i}{n-i+1}<t\,\middle|\,j,\,D(0)=n\right)$$
$$=\sum_{i=1}^{j}\prod_{\substack{k=1\\k\neq i}}^{j}\left(\frac{\frac{n-k+1}{j-k+1}}{\frac{n-k+1}{j-k+1}-\frac{n-i+1}{j-i+1}}\right)\left[1-\mathrm{e}^{-\mu\frac{n-i+1}{j-i+1}t}\right]. \tag{6.9}$$

Consider the following:

$$P[R<r|\,D(0),\,j]$$
$$=P\left(\frac{x}{T}<r\,\middle|\,D(0),\,j\right)$$
$$=\int_0^{\infty}P[x<rt|\,D(0),\,j,\,T=t]P(T=t)\,\mathrm{d}t\,. \tag{6.10}$$

Since we know that x will have an exponential distribution with hazard rate $[D(0)-j]\,\mu$, we obtain:

$$P[R<r|\,D(0)=n]$$
$$=1-\int_{t=0}^{\infty}\mathrm{e}^{-(n-j)\mu rt}\sum_{i=1}^{j}\prod_{k=1,k\neq i}^{j}\left(\frac{\frac{n-k+1}{j-k+1}}{\frac{n-k+1}{j-k+1}-\frac{n-i+1}{j-i+1}}\right)$$
$$\times\mu\frac{n-i+1}{j-i+1}\mathrm{e}^{-\mu\frac{n-i+1}{j-i+1}t}\,\mathrm{d}t$$
$$=1-\sum_{i=1}^{j}\frac{(n-i+1)}{[n-i+1+(j-i+1)(n-j)r]}$$
$$\times\prod_{k=1,k\neq i}^{j}\left(\frac{\frac{n-k+1}{j-k+1}}{\frac{n-k+1}{j-k+1}-\frac{n-i+1}{j-i+1}}\right). \tag{6.11}$$

This probability distribution is a pivotal since it does not depend on μ and can therefore be used to construct a prediction distribution for the time to detect the next fault (6.11), which can be simplified to give

$$P[R<r|\,D(0)=n]$$
$$=1-\frac{n!}{(n-j)!\,(n-j)^{j-1}}$$
$$\times\sum_{i=1}^{j}\left\{\frac{(j-i+1)^{j-1}(-1)^{i-1}}{[n-i+1+(j-i+1)(n-j)r]}\right.$$
$$\left.\times\frac{1}{(i-1)!(j-i)!}\right\}. \tag{6.12}$$

Taking the expectation with respect to $D(0)$, for which it is assumed we have a Poisson prior distribution, provides a CDF that can be used to obtain prediction intervals. The CDF in (6.13) for the ratio, i.e. R, is calculated assuming that there is at least one more fault remaining in the system design. The probability that all faults have been exposed given there has been j faults detected is expressed in (6.14)

$$P[R<r|\,D(0)\geq j+1,\,j]$$
$$=1-\left(\frac{\sum_{n=j+1}^{\infty}\frac{n!}{(n-j)!(n-j)^{j-1}}}{1-\sum_{k=0}^{j}\frac{\lambda^k\mathrm{e}^{-\lambda}}{k!}}\right.$$
$$\times\sum_{i=1}^{j}\left\{\frac{(j-i+1)^{j-1}\,(-1)^{i-1}}{[n-i+1+(j-i+1)\,(n-j)\,r]}\right.$$
$$\left.\left.\times\frac{1}{(i-1)!(j-i)!}\frac{\lambda^n\mathrm{e}^{-\lambda}}{n!}\right\}\right), \tag{6.13}$$

$$P(R=\infty|\,j)=\frac{\frac{\lambda^j\mathrm{e}^{-\lambda}}{j!}}{1-\sum_{k=0}^{j-1}\frac{\lambda^k\mathrm{e}^{-\lambda}}{k!}}. \tag{6.14}$$

Consider how this distribution changes as we expose all faults within the design. As the number of faults detected, j, increases towards $D(0)$, then the distribution of T approaches a gamma distribution as follows:

$$\lim_{j\to D(0)}T=\lim_{j\to D(0)}\sum_{i=1}^{j}t_i$$
$$=\lim_{j\to D(0)}\sum_{i=1}^{j}\frac{j-i+1}{D(0)-i+1}W_i=\sum_{i=1}^{j}W_i\,,$$

where W_i are independent and identically exponentially distributed and their sum has a gamma distribution with

parameters j and μ. Therefore, as j approaches $D(0)$, the distribution of R should approach the following:

$$P[R<r|\,D(0)=n,\,j] = \int_0^\infty P[x<rt|\,D(0)=n,\,j,\,T=t]\,P(T=t)\,\mathrm{d}t$$

$$= 1-\left(\frac{1}{1+r}\right)^j . \qquad (6.15)$$

Computationally, (6.15) will be easier to apply than (6.13), assuming that there exists at least one more fault in the design. In the following section the quality of (6.15) as an approximation to (6.13) will be evaluated.

6.3 Evaluation of Prediction Intervals for the Time to Detect Next Fault

In this section the CDF in (6.13) is investigated to assess how it is affected by parameter changes in λ, which represents the subjective input describing the expected number of faults within the design. The degree of closeness of the approximation of the simple distribution in (6.15) to the more computationally intensive distribution in (6.13) will be examined. Finally a table of key values for making predictions with this model are presented.

The model assumes that the time between the i-th and $(i+1)$-th fault detection is exponentially distributed with hazard rate $\mu[D(0)-i+1]$. Consequently, if $D(0)$ increases then the expected time to realise the next fault reduces. Therefore as λ, which is the expectation of $D(0)$, increases, the CDF for the ratio R shifts upwards. This is illustrated in Fig. 6.1, where the CDF in (6.13) is shown for various values of λ assuming there have been 1, 5, 10 and 25 faults detected.

Figure 6.2 illustrates the CDF in (6.13) assuming a mean number of faults of 1 and 25 and compares it to the asymptotic distribution (6.15). Interestingly the approximation improves when the number of faults observed is moderately lower than the mean, compared with when it is greater than the mean. However, convergence is slow. For example, there is a noticeable

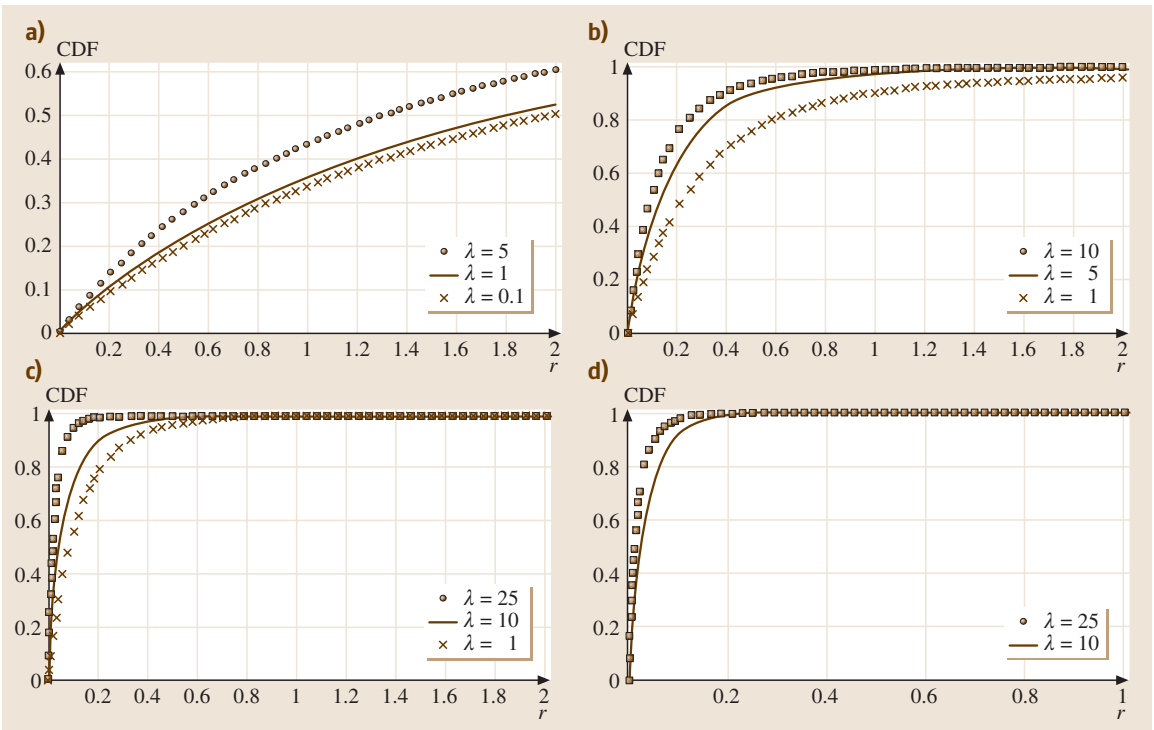


Fig. 6.1a–d Comparison of CDF for ratio R as λ changes; **(a)** 1 fault detected; **(b)** 5 faults detected; **(c)** 10 faults detected; **(d)** 25 faults detected

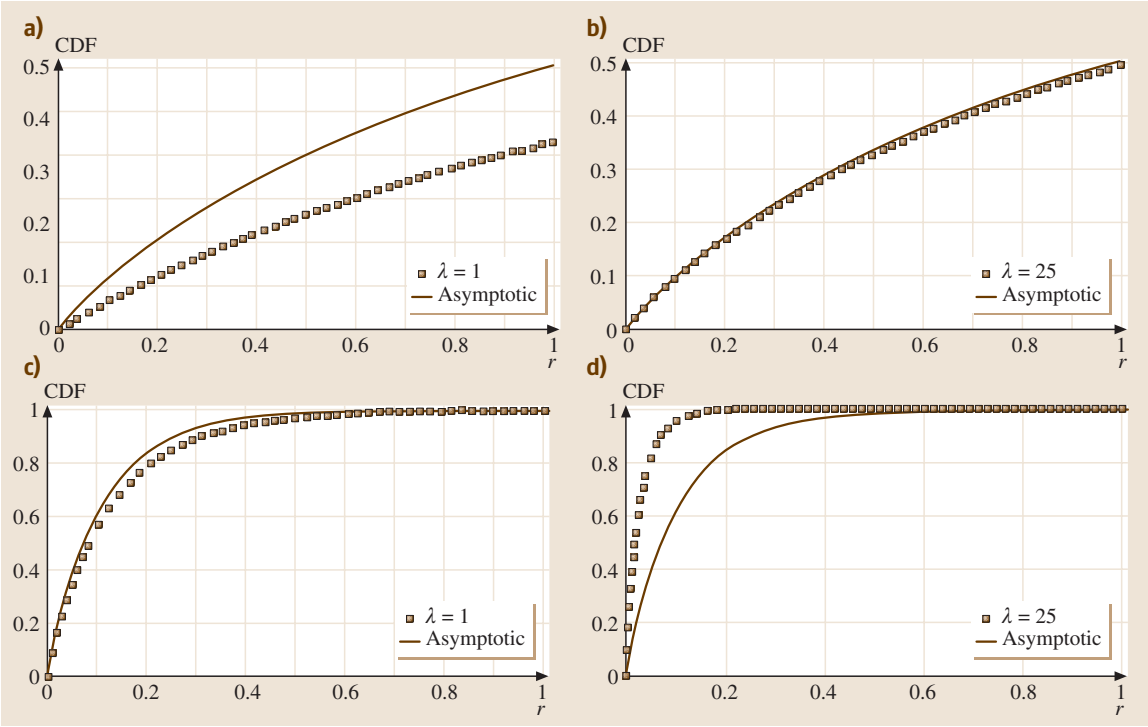


Fig. 6.2a–d Comparison of CDF for ratio R and asymptotic distribution; **(a)** 1 fault detected; **(b)** 1 fault detected; **(c)** 10 faults detected; **(d)** 10 faults detected

difference between the two CDFs in Fig. 6.2c, where there have been 10 faults exposed but the engineering experts expected only one.

The data in the plots in Figs. 6.1 and 6.2 was generated using Maple version 8, which has been used to support numerical computations. To avoid the need to reproduce calculations, summaries of the key characteristics of the CDF are presented in the following tables.

Table 6.1 provides a summary of the expectation of the ratio of the time to next fault detection and the sum of the preceding fault detection times. We consider the mean number of faults λ as it increases from 1 to 20 and the number of observed faults detected as it increases from 1 to 10. For the situation where there is one fault detected the mean is infinite. However, for the case of two or more faults detected the mean is finite. Therefore, the median may provide a more appropriate point estimate of a prediction if there is only one fault detected. In addition, the skewness of the distribution of the mean suggests that median estimators may be appropriate more generally. Note also that as both the number of faults detected increases and the mean number of faults within the design increases the differences in the mean of the distribution of the ratio decrease.

Table 6.1 Values of the mean of the distribution of R

$j\backslash\lambda$	1	5	10	15	20
1	∞	∞	∞	∞	∞
2	1.60	1.13	0.86	0.78	0.76
3	0.74	0.52	0.36	0.31	0.3
4	0.46	0.34	0.22	0.18	0.17
5	0.33	0.25	0.16	0.12	0.11
10	0.13	0.11	0.08	0.05	0.04

Table 6.2 Values of the median of the distribution of R

$j\backslash\lambda$	1	5	10	15	20
1	1.82	1.32	1.13	1.08	1.06
2	0.65	0.45	0.35	0.32	0.31
3	0.37	0.26	0.18	0.16	0.15
4	0.26	0.18	0.12	0.10	0.09
5	0.20	0.14	0.09	0.07	0.06
10	0.09	0.07	0.05	0.03	0.02

Table 6.3 Percentiles of the distribution of R

a) 90-th percentile					
$j\backslash\lambda$	1	5	10	15	20
1	16.48	11.97	10.17	9.7	9.5
2	3.46	2.44	1.86	1.7	1.64
3	1.7	1.22	0.84	0.73	0.69
4	1.09	0.83	0.52	0.42	0.39
5	0.79	0.6	0.38	0.29	0.25
10	0.32	0.27	0.2	0.13	0.09
b) 95-th percentile					
$j\backslash\lambda$	1	5	10	15	20
1	34.83	25.31	21.49	20.48	20.06
2	5.58	3.96	3	2.73	2.64
3	2.54	1.85	1.27	1.09	1.03
4	1.57	1.18	0.77	0.61	0.56
5	1.11	0.87	0.56	0.41	0.36
10	0.43	0.37	0.28	0.18	0.12
c) 99-th percentile					
$j\backslash\lambda$	1	5	10	15	20
1	181.61	132.03	112.14	106.73	104.54
2	14.53	10.42	7.83	7.15	6.9
3	5.45	4.04	2.76	2.35	2.22
4	3.07	2.38	1.56	1.21	1.16
5	2.07	1.67	1.10	0.78	0.68
10	0.72	0.65	0.52	0.36	0.22

Table 6.2 presents a summary of the median values from the distribution of R. For comparison the same values of j and λ have been chosen as in Table 6.1. The skew in the distribution is noticeable through the difference between the mean and the median. This difference decreases as the number of faults detected increases. The changes in the median are greater for smaller values of λ.

Table 6.3 presents summaries of the 90-th, 95-th and 99-th percentiles of the distribution of R. The skew is noticeable by the difference between the 95-th and 99-th percentile, where there is a larger difference for the situation where there have been fewer faults detected.

6.4 Illustrative Example

This example is based around the context and data from a reliability growth test of a complex electronic system. A desensitised version of the data is presented; however, this does not detract from the key issues arising through this reliability growth test and the way in which the data are treated. The aim of this example is to illustrate the process of implementing the modified IBM (MIBM) model for prediction and to reflect upon the strengths and weaknesses of this approach.

6.4.1 Construction of Predictions

A TAAF test regime has been used in this growth development test. The duration of the test was not determined a priori but was to be decided based upon the analysis of the test data. Two units have been used for testing. Both units have been tested simultaneously. The test conditions were such that they approximated the stress levels expected to be experienced during operation.

A prior distribution has been elicited before testing is commenced. The experts used to acquire this information are the engineers involved in the design and development of the system and specialists in specification of the test environment. The process for acquiring such a prior distribution is described in [6.11]. This process involves individual interviews with each expert, discussing novel aspects of the design, identifying engineering concerns and eliciting probabilities that measure their degree of belief that the concern will be realised as a fault during test. Group interviews are conducted where overlapping concerns are identified to explore correlation in the assessment. The probabilities are convoluted to obtain a prior distribution describing the number of faults that exist within the design. Typically a Poisson distribution provides a suitable form for this prior, and this is assumed within the MIBM model. For this illustration we use a Poisson distribution with a mean of 15.

The test was conducted for a total of 7713 test hours. This was obtained by combining the exposure on both test units. Nine faults were exposed during this period. Figure 6.3 illustrates the cumulative number of faults detected against test time. Visually, there is evidence of reliability growth, as the curve appears concave, implying that the time between fault detection is, on average, decreasing. There were two occasions where faults were detected in relatively short succession:

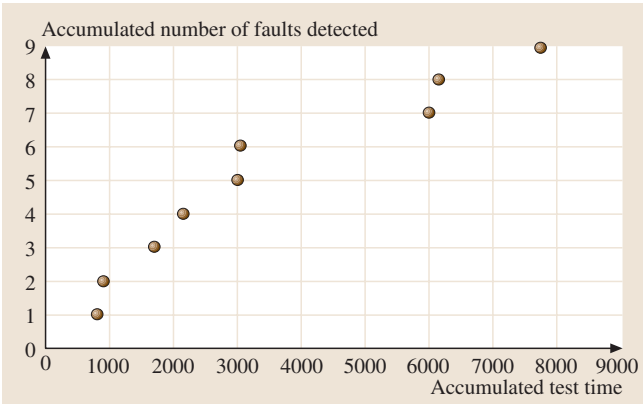


Fig. 6.3 Faults detected against accumulated test time

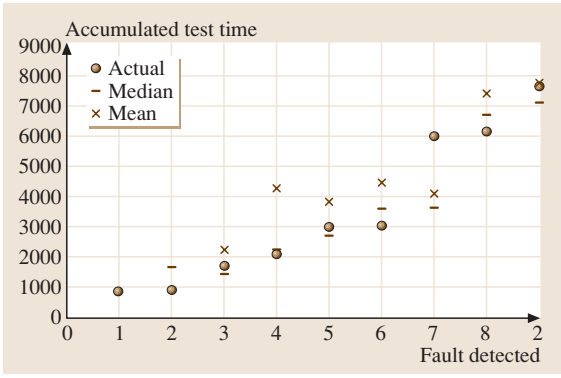


Fig. 6.4 Comparison between actual and model prediction fault detection times

this occurred between 3000 to 4000 hours and 6000 to 7000 hours.

For illustrative purposes we consider the realisations of faults sequentially and use the MIBM model to predict the time to the next fault detection. These predictions are conditioned on there being at least one more fault in the design at the time of prediction. The probability that all faults have been detected will also form part of the prediction.

Table 6.4 provides a summary of the estimates provided by the MIBM and these are also illustrated in Fig. 6.4. The median estimator appears to perform better than the mean. This is due to the large skew in the tail of the distribution of the predicted ratio resulting in a mean that is much greater than the median. All nine faults were observed at earlier times than the upper 95% prediction limit. The point estimates were poorest at the time of the seventh fault detection, which had been realised after an unusually long period of no fault detection.

Table 6.5 provides the point estimate of the number of faults remaining undetected within the design at each fault detection time. This is obtained using the formula (6.16), the derivation of which can be found in [6.12]. The MLE of μ was substituted in place of the parameter:

$$E\left[D\left(0\right)-N\left(t\right)|\,\mu\right]=\lambda\,\mathrm{e}^{-\mu t}\,. \tag{6.16}$$

Table 6.5 provides some confidence in the model we are using for this process. The expected number of faults remaining undetected decreases in line with the detection

Table 6.4 Predictions of fault detection times based on model

Fault number	Actual	Upper 95% prediction	Median prediction	Mean prediction
1	800			
2	900	17184	1664	
3	1700	5541	1444	2226
4	2150	5406	2244	4284
5	3000	5536	2705	3815
6	3050	7788	3599	4454
7	6000	6646	3630	4094
8	6150	10400	6704	7408
9	7713	11375	7100	7813

Table 6.5 Expected faults remaining undetected

Fault detected	Accumulated test time	$\hat{\mu}$	Expected faults remaining
1	800	8.31×10^{-5}	14.0
2	900	0.000148	13.1
3	1700	0.000125	12.1
4	2150	0.000135	11.2
5	3000	0.000127	10.2
6	3050	0.000147	9.6
7	6000	0.000108	7.8
8	6150	0.000116	7.3
9	7713	0.000112	6.3

Table 6.6 Probability of having detected all faults

Number of faults detected	Probability all faults have been detected
1	$4.588\,54\times10^{-6}$
2	$3.441\,42\times10^{-5}$
3	0.0002
4	0.0006
5	0.0019
6	0.0049
7	0.0105
8	0.0198
9	0.0337

of the faults and there is some stability in the estimation of μ.

The analysis conducted to obtain predictions assumes that there is at least one more fault remaining undetected within the design. However, we are assuming that there are a finite number of faults within the design described through the Poisson prior distribution with a mean of 15 at the start of test. Table 6.6 provides the probability that all faults have been detected at each fault detection times. This is a conditional probability calculated from the Poisson distribution. The formula used, which assumes j faults have been detected and a mean of λ, is provided in (6.17):

$$P\left[D\left(0\right)=j\,\middle|\,N\left(t'\right)=j\right]=\frac{\frac{\lambda^{j}\,\mathrm{e}^{-\lambda}}{j!}}{1-\sum_{k=0}^{j-1}\frac{\lambda^{k}\,\mathrm{e}^{-\lambda}}{k!}}\,. \quad (6.17)$$

The probability of having detected all faults increases as the number of faults increase, which is consistent with intuition. By the detection of the ninth fault there is still a small probability of all faults having been detected.

6.4.2 Diagnostic Analysis

Although visually the model appears to describe the variability within the data, the formal assessment of the validity of the CDF of the ratio R is assessed in this section.

Firstly, consider the number of fault detections where the time of detection was earlier than the median. This occurred four out of the eight times where a prediction was possible. Secondly, we compare the observed ratio of the time between successive fault detections and the sum of the proceeding fault detection times. Our hypothesis is that these observations have been generated from the CDF of R. Table 6.7 presents a summary of the observed ratio and the percentile to which the ratio corresponds in the CDF. Assuming that the model is appropriate then these observed percentiles should be consistent with eight observations being observed from a uniform distribution over the range [0, 1]. These percentiles appear to be uniformly distributed across [0, 1] with possibly a slight central tendency between 0.6 and 0.7.

Table 6.7 Observed ratios

Fault detected	Observed ratio	Percentile of ratio
1		
2	0.13	0.1
3	0.47	0.61
4	0.13	0.44
5	0.15	0.64
6	0.01	0.06
7	0.25	0.93
8	0.01	0.13
9	0.07	0.68

A formal test can be constructed to assess whether the percentiles in Table 6.7 are appropriately described as belonging to a uniform distribution. From *Bates* [6.17] the average of at least four uniform random variables is sufficient for convergence with the normal distribution. Therefore, the average of the eight uniform random variables in Table 6.7 is approximately normally distributed with a mean of 0.5 and a standard deviation of 0.029. The average of the observed percentiles is 0.449, which corresponds to a Z-score of -1.76, which is within the 95% bounds for a normal random variable and, as such, we conclude that these percentiles are uniformly distributed. Therefore, the model cannot be rejected as providing an appropriate prediction of the variability of the time to the next fault detection within the test.

6.4.3 Sensitivity with Respect to the Expected Number of Faults

The predictions require a subjective input describing the expected number of faults within the design and as such it is worth exploring the impact of a more pessimistic or optimistic group of experts.

From Sect. 6.3 it is known that the CDF of the ratio R is most sensitive to changes in λ about the median and the impact is lower for the upper prediction limits. The error is defined as the observed minus the predicted value. A summary of selected performance measures for the error corresponding to various values of λ, specifically a pessimistic estimate of 20 and an optimistic

Table 6.8 Prediction errors

	$\lambda = 10$	$\lambda = 15$	$\lambda = 20$
Error (median)	9	197	−612
Error (mean)	−911	−619	−821
MAD (median)	657	687	612
MAD (mean)	1323	1163	1053

estimate of 10 have been computed and are summarised in Table 6.8. The average error is smallest for the optimistic estimate using the median to predict, but greatest for the optimistic estimate using the mean to predict. The mean absolute deviation (MAD) has also been calculated as a method of assessing accuracy, but there is little difference between these values.

6.4.4 Predicting In-Service Failure Times

Accessing in-service data is much more challenging than obtaining data from a controlled reliability growth test. Consequently, assessing the quality of the predictions for in-service performance cannot be expected to be as thorough. However, for this system summary statistics have been obtained from the first two years of operation. The observed Mean Time Between Failure (MTBF) was 1610 h at 18 mon, 3657 h after 19 mon and 1876 h after 22 mon of operation. The model predicts an MTBF of 1888 h, assuming λ is 15. The optimistic estimate of λ of 10 results in an estimate of 1258 h, while the pessimistic estimate of 20 results in an MTBF of 2832 h.

6.5 Conclusions and Reflections

A simple model for predicting the time of fault detection on a reliability growth test with small sample sizes has been described. The model requires subjective input from relevant engineers describing the number of faults that may exist within the design and estimates the rate of detection of faults based on a mixture of the empirical test data and the aforementioned expert judgement.

An illustrative example has been constructed based on a desensitised version of real data where the model has been used in anger. Through this example the processes of constructing the predictions and validating the model have been illustrated.

Despite their complexity, modern electronic systems can be extremely reliable and hence reliability growth tests are increasingly being viewed as a luxury that industry can no longer afford. Obviously, omitting reliability growth tests from development programmes would not be sensible for systems with complex interactions between subassemblies, since potentially much useful information could be gained to inform development decisions. This reinforces that there remains a need to model the fault detection process to provide appropriate decision support. The modelling framework considered here lends itself easily to test data on partitions of the system, as the key drivers are the number of faults in the design and the rate at which there are detected. These are easily amalgamated to provide an overall prediction of the time until next system failure.

The approach considered here combines both Bayesian and frequentist approaches. The main reason for this is that, in our experience, we were much more confident of the subjective data obtained describing the number of faults that may exist within a design and less so about subjective assessments describing when these would realised.

The recent paradigm shift in industry to invest more in reliability enhancement in design and early development means that observable failures have decreased and hence presented new growth modelling challenges. Analysis during product development programmes is likely to become increasingly dependent upon engineering judgement, as the lack of empirical data will result in frequentist techniques yielding uninformative support measures. Therefore, Bayesian methods will become a more realistic practical approach to reliability growth modelling for situations where engineering judgement exists. However, Bayesian models will only be as good as the subjective judgement used as input.

References

6.1 N. Rosner: *System Analysis—Nonlinear Estimation Techniques*, Proc. Seventh National Symposium on Reliability, Quality Control (Institute of Radio Engineers, New York 1961)

6.2 J. M. Cozzolino: Probabilistic models of decreasing failure rate processes, Naval Res. Logist. Quart. **15**, 361–374 (1968)

6.3 Z. Jelinski, P. Moranda: Software reliability research. In: *Statistical Computer Performance Evaluation*, ed. by W. Freiberger (Academic, New York 1972) pp. 485–502

6.4 M. Xie: Software reliability models—A selected annotated bibliography, Soft. Test. Verif. Reliab. **3**, 3–28 (1993)

6.5 E. H. Forman, N. D. Singpurwalla: An empirical stopping rule for debugging and testing computer software, J. Am. Stat. Assoc. **72**, 750–757 (1977)

6.6 B. Littlewood, J. L. Verrall: A Bayesian reliability growth model for computer software, Appl. Stat. **22**, 332–346 (1973)

6.7 R. Meinhold, N. D. Singpurwalla: Bayesian analysis of a commonly used model for describing software failures, The Statistician **32**, 168–173 (1983)

6.8 J. Quigley, L. Walls: Measuring the effectiveness of reliability growth testing, Qual. Reliab. Eng. **15**, 87–93 (1999)

6.9 L. H. Crow: *Reliability analysis of complex repairable systems reliability and Biometry*, ed. by F. Proschan, R. J. Serfling (SIAM, Philadelphia 1974)

6.10 J. T. Duane: Learning curve approach to reliability monitoring, IEEE Trans. Aerosp. **2**, 563–566 (1964)

6.11 L. Walls, J. Quigley: Building prior distributions to support Bayesian reliability growth modelling using expert judgement, Reliab. Eng. Syst. Saf. **74**, 117–128 (2001)

6.12 J. Quigley, L. Walls: Confidence intervals for reliability growth models with small sample sizes, IEEE Trans. Reliab. **52**, 257–262 (2003)

6.13 J. Quigley, L. Walls: Cost–benefit modelling for reliability growth, J. Oper. Res. Soc. **54**, 1234–124 (2003)

6.14 L. Walls, J. Quigley, M. Kraisch: Comparison of two models for managing reliability growth during product development, IMA J. Math. Appl. Ind. Bus. (2005) (in press)

6.15 H. A. David, H. N. Nagaraja: *Order Statistics*, 3rd edn. (Wiley, New York 2003)

6.16 I. J. Good: On the weighted combination of significance tests, J. R. Stat. Soc. Ser. B **17**, 264–265 (1955)

6.17 G. E. Bates: Joint distributions of time intervals for the occurrence of successive accidents in a generalised Polya scheme, Ann. Math. Stat. **26**, 705–720 (1955)

7. Promotional Warranty Policies: Analysis and Perspectives

Warranty is a topic that has been studied extensively by different disciplines including engineering, economics, management science, accounting, and marketing researchers [7.1, p. 47]. This chapter aims to provide an overview on warranties, focusing on the cost and benefit perspective of warranty issuers.

After a brief introduction of the current status of warranty research, the second part of this chapter classifies various existing and several new promotional warranty policies to extend the taxonomy initiated by *Blischke* and *Murthy* [7.2].

Focusing on the quantitative modeling perspective of both the cost and benefit analyses of warranties, we summarize five problems that are essential to warranty issuers. These problems are: i) what are the warranty cost factors; ii) how to compare different warranty policies; iii) how to analyze the warranty cost of multi-component systems; iv) how to evaluate the warranty benefits; v) how to determine the optimal warranty policy.

A list of future warranty research topics are presented in the last part of this chapter. We hope that this will stimulate further interest among researchers and practitioners.

Warranty is an obligation attached to products (items or systems) that requires the warranty issuers (manufacturers or suppliers) to provide compensation to consumers according to the warranty terms when the warranted products fail to perform their pre-specified functions under normal usage within the warranty coverage period. Similar definitions can be found in *Blischke* and *Murthy* [7.1, 3], *McGuire* [7.4], and *Singpurwalla* and *Wilson* et al. [7.5]. Based on this definition, a warranty contract should contain at least three characteristics: the coverage period (fixed or random), the method of compensations, and the conditions under which such compensations would be offered. The last characteristic is closely related to warranty execution since it clarifies consumers' rights and protects warranty issuers from excessive false claims. From the costing perspective, the first two characteristics are more important to manufacturers because they determine the depth of the protection against premature failures and the direct cost related to those failures.

Traditionally, warranty serves as a protection instrument attached to products sold to consumers. There are two facets of the protection role: on one hand, it guarantees a properly functioning product for at least a period of w, either financially or physically. On the other hand, it also specifies an upper bound on the liability of the supplier induced by the warranty. In addition to the protection role, warranty has always been one of the most important elements in business marketing strategy. As indicated in [7.4, p.1], manufacturers' primary rationale for offering warranty is to support their products to gain some advantage in the market, either by expressing the company's faith in the product quality, or by competing with other firms. Due to the more than ever fierce competition in the modern economy, the market promotion role of warranty has become even more significant. Manufacturers are fighting with each other through various

channels from competitive pricing, improved product reliability, to more comprehensive warranties. Because of technology constraints or time constraint, it is usually difficult to improve product quality in a short time. As a result, warranty has evolved as an essential part of marketing strategy, along with pricing and advertising, which is especially powerful during the introduction period of new, expensive products such as automobiles and complex machinery.

In the last two decades, warranty has been studied extensively among many disciplines such as engineering, economics, statistics, marketing and management science, to name a few. Consequently, the literature on warranty is not only vast, but also disjoint [7.1]. There are three books and hundreds of journal articles that have addressed warranty-related problems within the last ten years. A comprehensive collection of related references up to 1996 can be found in [7.3]. In general, researchers in engineering are interested in quality control and improving product reliability to reduce production and service costs. Some of the major references are *Chen* et al. [7.6], *Djamaludin* et al. [7.7], *Hedge* and *Kubat* [7.8], *Mi* [7.9], *Murthy* and *Hussain* [7.10], *Nguyen* and *Murthy* [7.11], and *Sahin* [7.12]. Economists usually treat warranty as a special type of insurance. Consequently, they developed the economic theory of warranties as one of many applications of microeconomics. We refer readers to *DeCroix* [7.13], *Emons* [7.14, 15], *Lutz* and *Padmanabhan* [7.16], *Padmanabhan* and *Rao* [7.17], *Murthy* and *Asgharizadeh* [7.18] and the references therein. Statisticians mainly focus on warranty claim prediction, statistical inference of warranty cost, and estimation of product reliability or availability. Some of the key references are *Frees* [7.19, 20], *Ja* et al. [7.21], *Kalbfleisch* [7.22], *Kao* and *Smith* [7.23, 24], *Menzefricke* [7.25], *Padmanabhan* and *Worm* [7.26] and *Polatoglu* [7.27]. A long-term trend in warranty study is the focus on various warranty-management aspects. Some recent references are *Chun* and *Tang* [7.28], *Ja* et al. [7.21], *Lam* and *Lam* [7.29], *Wang* and *Sheu* [7.30], and *Yeh* et al. [7.31, 32]. *Blischke* and *Murthy* [7.33] developed a framework for the analytical study of various issues related to warranty. Recently, *Murthy* and *Djamaludin* [7.34] enriched the framework by summarizing the literature since 1992 from an overall business perspective. Another review by *Thomas* and *Rao* [7.35] provided some suggestions for expanding the analysis methods for making warranty decisions.

In this chapter, we briefly review some recent work in warranty literature from the manufacturers' perspective. The objectives of this chapter are to classify various existing and relatively new warranty policies to extend the taxonomy proposed in [7.2], and to summarize and illustrate some fundamental warranty economic problems.

7.1 Classification of Warranty Policies

Numerous warranty policies have been studied in the last several decades. *Blischke* and *Murthy* [7.2] presented a taxonomy of more than 18 warranty policies and provided a precise statement of each of them. In this section, we extend the taxonomy by addressing several recently proposed policies that might be of interests to warranty managers. It should be noted that we mainly focus on type *A* policies [7.2], which, based on the taxonomy, are referred to as policies for single items and not involving product development.

7.1.1 Renewable and Nonrenewable Warranties

One of the basic characteristics of warranties is whether they are renewable or not. For a regular renewable policy with warranty period w, whenever a product fails within w, the buyer is compensated according to the terms of the warranty contract and the warranty policy is renewed for another period w. As a result, a warranty cycle T, starting from the date of sale, ending at the warranty expiration date, is a random variable whose value depends on w, the total number of failures under the warranty, and the actual failure inter-arrival times. Renewable warranties are often offered for inexpensive, nonrepairable consumer electronic products such as microwaves, coffee makers, and so forth, either implicitly or explicitly. One should notice that theoretically the warranty cycle for a renewable policy can be arbitrarily large. For example, consumers can induce the failures so that they keep on getting new warranties indefinitely. Such moral hazard problems might be one of the reasons that renewable policies are not as popular as nonrenewable ones among warranty issuers.

One way to remedy this problem is to modify the regular renewable policy in the following way: instead of offering the original warranty with a period of w repeatedly upon each renewing, warranty issuers

could set $w_i = \alpha w_{i-1}$, $\alpha \in (0, 1]$, for $i = 1, 2, \cdots$, where w_i is the warranty length for the i-th renewing, and $w_0 = w$. Actually, this defines a new type of renewable warranty, which we refer to as geometric renewable warranty policies. Clearly, a geometric renewable policy is a generalization of a regular renewable policy, which degenerates to the latter when $\alpha = 1$.

The majority of warranties in the market are nonrenewable; for these the warranty cycle, which is the same as the warranty period, is not random, but predetermined (fixed), since the warranty obligation will be terminated as soon as w units of time pass after sale. This type of policies is also known as a fixed-period warranty.

7.1.2 FRW, FRPW, PRW, CMW, and FSW Policies

According to the methods of compensation specified in a warranty contract upon premature failures, there are three basic types of warranties: free replacement warranty (FRW), free repair warranty (FRPW), and pro-rata warranty (PRW). Combination warranty (CMW) contains both features of FRW/FRPW and PRW. Full-service warranty, (FSW), which is also known as preventive maintenance warranty, is a policy that may be offered for expensive deteriorating complex products such as automobiles. Under this type of policies, consumers not only receive free repairs upon premature failures, but also free (preventive) maintenance.

For nonrepairable products, the failed products under warranty will usually be replaced free of charge to consumers. Such a policy is often referred to as a free replacement warranty or an unlimited warranty. In practice, even if a product is technically repairable, sometimes it will be replaced upon failure since repair may not be economically sound. As a result, for inexpensive repairable products, warranty issuers could simply offer FRW policies. Consequently, these inexpensive repairable products can be treated as nonrepairable. However, for repairable products, if the warranty terms specify that, upon a valid warranty claim, the warranty issuer will repair the failed product to working condition free of charge to buyers, then such a policy is a so-called free repair warranty. In practice, it is not rare that a warranty contract specifies that the warranty issuer would repair or replace a defective product under certain conditions. This is the reason why most researchers do not treat FRW and FRPW separately. Nevertheless, we feel that it is necessary to differentiate these two type of policies based on the following reasoning: first, repair cost is usually much lower than replacement cost except for inexpensive products; secondly, by clearly defining the compensation terms, warranty issuers may establish a better image among consumers, which can surely be helpful for the marketing purpose.

Under a FRW policy, since every failed product within T is replaced by a new one, it is reasonable to model all the subsequent failure times by a single probability distribution. However, under a FRPW, it is necessary to model the repair impact on failure times of a warranted product. If it is assumed that any repair is as-good-as-new (perfect repair), then from the modeling perspective, there is little difference between FRW and FRPW. For deteriorating complex systems, minimal repair is a commonly used assumption. Under this assumption, a repair action restores the system's failure rate to the level at the time epoch when the last failure happened. Minimal repair was first introduced by *Barlow* and *Proschan* [7.36]. Changing a broken fan belt on an engine is a good example of minimal repair since the overall failure rate of the car is nearly unchanged. Perfect repair and minimal repair represent two extremes relating to the degree of repair. Realistically, a repair usually makes a system neither as-good-as-new, nor as-bad-as-old (minimal repair), but to a level in between. This type of repair is often referred to as imperfect repair. In the literature of maintenance and reliability, many researchers have studied various maintenance policies considering imperfect repair. A recent review on imperfect maintenance was given by *Pham* and *Wang* [7.37]. In the warranty literature, the majority of researchers consider repairs as either perfect or minimal. Little has been done on warranty cost analysis considering imperfect repair.

Both FRW and FRPW policies provide full coverage to consumers in case of product failures within T. In contrast, a PRW policy requires that buyers pay a proportion of the warranty service cost upon a failure within T in exchange for the warranty service such as repair or replacement, cash rebate or a discount on purchasing a new product. The amount that a buyer should pay is usually an increasing function of the product age (duration after the sale). As an example, suppose the average repair/replacement cost per failure is c_s, which could be interpreted as the seller's cost per product without warranty, if a linear pro-rata function is used, then the cost for a buyer upon a failure at time t, $t < w$, is $c_s \frac{t}{w}$. The corresponding warranty cost incurred to the manufacturer is $c_s\left(1 - \frac{t}{w}\right)$. PRW policies are usually renewable and are offered for relatively inexpensive products such as tires, batteries, and so forth.

Generally speaking, FRW and FRPW policies are in the favor of buyers since manufacturers take all the re-

sponsibility of providing products that function properly during the whole warranty cycle [7.1, p. 221]. In other words, it is the manufacturers that bear all the warranty cost risk. In contrast, for PRW policies manufacturers have the relative advantage with regard to the warranty cost risk. Although they do have to offer cash rebates or discounts to consumers if failures happen during T, they are usually better off no matter what consumers choose to do. If a consumer decides not to file a warranty claim, then the manufacturer saves himself the cash rebate or other type of warranty service. If instead a warranty claim is filed, the manufacturer might enjoy the increase in sales or at least the warranty service cost is shared by the consumer.

To balance the benefits between buyers and sellers, a combination warranty (CMW) that contains both features of FRW/FRPW and PRW policies was created. CMW is a policy that usually includes two warranty periods: a free repair/replacement period w_1 followed by a pro-rata period w_2. This type of warranties is not rare today because it has significant promotional value to sellers while at the same time it provides adequate control over the costs for both buyers and sellers [7.3, p. 12].

For deteriorating complex products, it is essential to perform preventive maintenance to achieve satisfactory reliability performance. Maintenance involves planned and unplanned actions carried out to retain a system at, or restore it to, an acceptable operating condition [7.38]. Planned maintenance is usually referred to as preventive maintenance while unplanned maintenance is labeled as corrective maintenance or repair. The burden of maintenance is usually on the consumers' side. In [7.39], *Bai* and *Pham* proposed a renewable full-service warranty for multi-component systems under which the failed component(s) or subsystem(s) will be replaced; in addition, a (preventive) maintenance action will be performed to reduce the chance of future product failures, both free of charge to consumers. They argue that such a policy is desirable for both consumers and manufacturers since consumers receive better warranty service compared to traditional FRPW policies, while at the same time manufacturers may enjoy cost savings due to the improved product reliability by the maintenance actions. By assuming perfect maintenance, they derived the probability distributions and the first two moments of the warranty cost per warranty cycle for series, parallel, series–parallel, and parallel–series systems.

Many researchers have studied warranty-maintenance problems. Among them Chun [7.40] determined the optimal number of periodic maintenance actions during the warranty period by minimizing the expected warranty cost (EWC). *Jack* and *Dagunar* [7.41] generalized Chun's idea by considering unequal preventive maintenance intervals. *Yeh* [7.32] further extended the work by including the degree of maintenance as one of the decision variables along with the number of maintenance actions and the maintenance schedule. All of these three researches aim to obtain the optimal maintenance warranty to assist manufacturers' decision-making. A related problem is the determination of the optimal maintenance strategy following the expiration of warranty from the consumers' perspective. *Dagpunar* and *Jack* [7.42] studied the problem by assuming minimal repair. Through a general approach, *Sahin* and *Polatoglu* [7.43] discussed both stationary and non-stationary maintenance strategies following the expiration of warranty. They proved the pseudo-convex property of the cost rate function under some mild conditions.

7.1.3 Repair-Limit Warranty

In maintenance literature, many researchers studied maintenance policies set up in such a way that different maintenance actions may take place depending on whether or not some pre-specified limits are met. Three types of limits are usually considered: repair-number-limit, repair-time-limit, and repair-cost-limit. Those maintenance policies are summarized by *Wang* [7.44].

Similarly, three types of repair-limit warranties may be considered by manufacturers: repair-number-limit warranty (RNLW), repair-time-limit warranty (RTLW), and repair-cost-limit warranty (RCLW). Under a RNLW, the manufacturer agrees to repair a warranted product up to m times within a period of w. If there are more than m failures within w, the failed product shall be replaced instead of being repaired again. *Bai* and *Pham* [7.45] recently studied the policy under the imperfect-repair assumption. They derived the analytical expressions for the expected value and the variance of warranty cost per product sold through a truncated quasi-renewal-process approach.

AN RTLW policy specifies that, within a warranty cycle T, any failures shall be repaired by the manufacturer, free of charge to consumers. If a warranty service cannot be completed within τ unit of time, then a penalty cost occurs to the manufacturer to compensate the inconvenience of the consumer. This policy was analyzed by *Murthy* and *Asgharizadeh* [7.18] in the context of maintenance service operation.

For a RCLW policy, there is a repair cost limit τ in addition to an ordinary FRPW policy. That is, upon each

failure within the warranty cycle T, if the estimated repair cost is greater than τ, then replacement instead of repair shall be provided to the consumer; otherwise, normal repair will be performed. This policy has been studied by *Nguyen* and *Murthy* [7.46] and others.

It should be noted that various repair limits as well as other warranty characteristics such as renewing may be combined together to define a new complex warranty. For example, it is possible to have a renewable repair-time-limit warranty for complex systems. Such combinations define a large set of new warranty policies that may appear in the market in the near future. Further study is needed to explore the statistical behavior of warranty costs of such policies to assist decisions of both manufacturers and consumers.

7.1.4 One-Attribute Warranty and Two-Attribute Warranty

Most warranties in practice are one-attribute, for which the warranty terms are based on product age or product usage, but not both. Compared to one-attribute warranties, two-attribute warranties are more complex since the warranty obligation depends on both the product age and product usage as well as the potential interaction between them. Two-attribute warranties are often seen in automobile industry. For example, Huyndai, the Korean automobile company, is currently offering 10 years/100 000 miles limited FRPW on the powertrain for most of their new models.

One may classify two-attribute warranties according to the shape of warranty coverage region. *Murthy* et al. defined four types of two-attribute warranties labeled as policy A to policy D (Fig. 1 in [7.47]). The shapes of the warranty regions are rectangular, L-shaped with no limits on age or usage, L-shaped with upper limits on age and usage, and triangular, respectively. Based on the concept of the iso-cost curve, *Chun* and *Tang* [7.28] proposed a set of two-attribute warranty policies for which the expected present values of future repair costs are the same. Some other plausible warranty regions for two-attribute warranty policies were discussed by *Singpurwalla* and *Wilson* [7.5].

In general, there are two approaches in the analysis of two-attribute warranties, namely, the one-dimensional (1-D) approach and the two-dimensional (2-D) approach. The 1-D approach assumes a relationship between product age and usage; therefore it eventually converts a two-attribute warranty into a corresponding one-attribute warranty. This approach is used by *Moskowitz* and *Chun* [7.48], and *Chun* and *Tang* [7.28]. The 2-D approach does not impose a deterministic relationship between age and usage. Instead, a bivariate probability distribution is employed for the two warranty attributes. *Murthy* et al. [7.47] followed the idea and derived the expressions for the expected warranty cost per item sold and for the expected life cycle cost based on a two-dimensional renewal processes. *Kim* and *Rao* [7.49] obtained the analytical expressions for the warranty cost for the policies A and B defined in [7.47] by considering a bivariate exponential distribution. Perhaps the most comprehensive study of two-attribute warranties so far is by *Singpurwalla* and *Wilson* [7.5], in which, through a game-theory set up, they discussed in detail both the optimum price-warranty problem and the warranty reserve determination problem.

7.2 Evaluation of Warranty Policies

Two phenomena make the study of warranties important. First, warranty has become common practice for manufacturers. According to the survey conducted by *McGuire*, nearly 95% percent of producers of industrial products provide warranties on all of their product lines [7.4, p. 1]; secondly, there is a huge amount of money involved in warranty programs. Based on a report by the Society of Mechanical Engineering (www.sme.org), the annual warranty cost is about 6 billion dollars for Ford, General Motors and Chrysler combined in the year 2001.

Among many issues related to warranty, there are two fundamental questions that must be answered, especially for warranty issuers: (1) how much a warranty will cost; (2) how much benefit can be earned from a certain warranty. This section summarizes some ideas and discussions appeared in the literature that are closely related to these two questions.

7.2.1 Warranty Cost Factors

Due to the random nature of many warranty cost factors such as product failure times, warranty cost is also a random variable whose statistical behavior can be determined by establishing mathematical links between warranty factors and warranty cost. There are numerous factors that may be considered in warranty

studies. Among them, we believe that the followings are of great importance: the characteristics of warranty policies; warranty service cost per failure; product failure mechanism; impact of warranty service on product reliability; warranty service time; and warranty-claim-related factors.

Different warranty policies may require different mathematical models for warranty cost. One way to model the warranty cost per item sold is through a stochastic counting process $[N(t), t \geq 0]$, which represents the number of failures over time of a warranted product. Let $S_1, S_2, \cdots$ be the subsequent failure times, and denote by C_{S_i} the warranty cost associated with the i-th failure. Assuming that all product failures are claimed, that all claims are valid, and instant warranty service, then the total warranty cost per item sold, $C(w)$, can be expressed as

$$C(w) = \begin{cases} \sum_{i=0}^{N[T(w)]} C_{S_i}, & \text{for } N[T(w)] = 1, 2, \cdots \\ 0, & \text{for } N[T(w)] = 0\,. \end{cases} \tag{7.1}$$

From (7.1), it is clear that the probabilistic behavior of $C(w)$ solely depends on $N[T(w)]$ (the number of failures within a warranty cycle T) and C_{S_i}, as well as the potential interaction between them. In general it is very difficult to determine the distribution of $C(w)$. However, it is possible to obtain the moments of $C(w)$ using modern stochastic process theory and probability theory.

For nonrepairable products or repairable products with a single component, warranty service cost per failure is often assumed to be constant. However, for repairable multi-component products, warranty service cost per failure in general is a random variable whose distribution is related to the product (system) structure and the warranty service cost for each component.

Product (system) failure mechanism can be described by the distributions of subsequent system failure times. This involves the consideration of system structure, the reliability of components and the impact of repair on components' reliability and system reliability. System structure is essential in determining system reliability. Extensive research on reliability modeling has been done for different systems such as series–parallel systems, parallel–series systems, standby systems, k-out-of-n systems, and so forth, in the literature of reliability [7.50]. Unfortunately, to our knowledge, there is no complete theory or methodology in warranty that incorporates the consideration of various system structure.

If a warranted product is nonrepairable or the as-good-as-new repair assumption is used for repairable products, then a single failure-time distribution can be adopted to describe the subsequent product failure times under warranty. However, if a warranted product is repairable and repairs are not as-good-as-new, then the failure time distribution(s) of repaired products differ(s) from that of a new product. This situation may be modeled by considering a failure-time distribution for all repaired products different from that of new products [7.1]. Strictly speaking, distributions of subsequent failure times of a repairable product are distinct, therefore, such an approach can be only viewed as an approximation.

As mentioned in Sect. 7.1, warranty compensation includes free replacement, free repair or cash rebate. For the case of free replacement, warranty service cost per failure for manufacturers is simply a constant that does not depend on the product failure times. In the case of cash rebate (pro-rata policy), warranty cost per failure usually relies on product failure time as well as the rebate function. When repair, especially the not as-good-as-new repair, is involved in warranty service, one has to model the repair impact on product reliability, which in turn has a great impact on warranty cost per failure. One way to model subsequent failure times under this situation is to consider them as a stochastic process. Consequently, modern stochastic theory of renewal processes, nonhomogeneous Poisson processes, quasi-renewal processes [7.38] and general point processes could be applied.

To our knowledge, most warranty literature assumes that warranty service is instant. This may be justified when the warranty service time is small compared to the warranty period or the warranty cycle. A better model is to incorporate explicitly the service times into warranty cost modeling. One recent attempt to include non-zero service time in warranty analysis is by *Murthy* and *Asgharizadeh* [7.18]. In this chapter, they developed a game-theoretic formulation to obtain the optimal decision in a maintenance service operation.

Warranty claims-related factors include the response of consumers to product failures and the validation of warranty claims by warranty issuers. It is no secret that not all consumers will make warranty claims even if they are entitled to do so. It is also true that warranty issuers, to serve their own benefits, usually have a formal procedure to validate warranty claims before honoring them. Such situations may be modeled by assigning two new parameters α and β, where α is the probability that

a consumer will file a claim upon a failure within T, and β is the proportion of the rejected claims [7.51].

There are other factors that may be of importance in warranty cost evaluation such as nonconforming product quality [7.6], multiple modes of failure, censored observations [7.20], and etc. Unfortunately, it is impossible to consider all the factors in one warranty cost model. Even if such a model exists, it would be too complicated to be applied.

7.2.2 Criteria for Comparison of Warranties

Warranty managers usually have several choices among various warranty policies that might be applied to a certain type of products. This requires some basic measures as the criteria to make the comparison among these policies.

There are several measures available, including expected warranty cost (EWC) per product sold, expected discounted warranty cost (EDWC) per warranty cycle, monetary utility function and weighted objective function. EWC and EDWC are more popular than the others since they are easy to understand and can be estimated relatively easily. The key difference between them is that the latter one considers the value of time, an important factor for warranty cost accounting and financial managers.

To our opinion, monetary utility function, $U(x)$, is a better candidate for the purpose of comparing warranty policies. The functional form of $U(x)$ reflects the manufacturer's risk attitude. If a manufacturer is risk-neutral, then $U(x)$ is linear in x. This implies that maximizing $\mathrm{E}[U(x)]$ is the same as maximizing $U[\mathrm{E}(x)]$. However, manufacturers may be risk-averse if they are concerned about the variations in profit or in warranty cost. For example, a particular manufacturer may prefer a warranty with less cost variation than another with much larger variation in warranty cost if the difference between the EWCs is small. If this is the case, then it can be shown that the corresponding utility function is concave [7.52]. The main difficulty of the utility theory approach is that utility functions are subjective.

Weighted objective functions could also be used for the purpose of comparing warranties for manufacturers. One commonly used weighted objective function is $\mathrm{E}[\pi(x)] - \rho \mathrm{V}[\pi(x)]$, where ρ is a positive parameter representing the subjective relative importance of the risk (variance or standard deviation) against the expectation and $\pi(x)$ is the manufacturers profit for a given warranty policy x. Interestingly, such an objective function coincides to a special case of the utility theory approach when the manufacturer's subjective utility function is assumed to only depend on the first two centered moments of $\pi(x)$ [7.53, 54].

In the above discussion, the term *warranty cost* refers to the manufacturer's cost per warranted product. In our opinion, this is the fundamental measure for the purpose of evaluating any warranty for manufacturers since it provides precise information on the additional cost incurred to manufacturers due to warranty. An equally useful measure is the discounted warranty cost (DWC) per cycle. This measure incorporates the value of time, therefore it is useful when warranty managers are interested in determining warranty reserve level. It is also of importance to financial managers performing warranty cost analysis.

Some researchers have proposed warranty cost per unit time, or warranty cost rate, as the primary warranty cost measure. As indicated by *Blischke* and *Murthy* [7.3], warranty cost rate is useful in managing warranty servicing resources, such as parts inventory over time with dynamic sales.

Another related measure is warranty cost over a product life cycle. *Blischke* and *Murthy* named this cost as *life cycle cost-II* (LCC-II) [7.1]. A product life cycle begins with the launch of the product onto the market and ends when it is withdrawn.

For consumers, warranty cost analysis is usually conducted over the life time of a product. In [7.1], this cost is labeled as *life cycle cost-I* (LCC-I). LCC-I is a consumer-oriented cost measure and it includes elements such as purchase cost, maintenance and repair costs following expiration of a warranty, operating costs as well as disposal costs.

7.2.3 Warranty Cost Evaluation for Complex Systems

Most products (systems), especially expensive ones, are composed of several nonrepairable components. Upon a failure, the common repair practice is to replace failed components instead of replacing the whole system. For such products, warranty may be offered for each of the individual components, or for the whole system. For the former case, the warranty cost modeling and analysis for single-component products can be applied readily. In fact, most warranty literature focuses on the analysis of warranty for single-component systems via a black-box approach. However, for the latter case, it is necessary to investigate warranty with explicit consideration of system structure because evidently system structure has

a huge impact on product reliability, therefore it is a crucial factor in warranty cost study. Unfortunately, as indicated by *Chukova* and *Dimitrov* [7.55, pp. 544], so far there has been only limited study on this topic.

Some researchers have discussed the warranty cost modeling for parallel systems. For example, *Ritchken* [7.56] provided an example of a two-component parallel system under a two-dimensional warranty. *Hussain* and *Murthy* [7.57] also discussed warranty cost estimation for parallel systems under the setting that uncertain quality of new products may be a concern for the design of warranty programs. *Chukova* and *Dimitrov* [7.55] presented a two-component parallel system under a FRPW policy. Actually, for nonrepairable parallel systems, the modeling techniques of warranty cost is essentially the same as that of black-box systems unless the system is considered as repairable.

To our knowledge, the only published work about warranty study on series systems is by *Chukova* and *Dimitrov* [7.55, p. 579–580]. They derived the EWC per system sold for a two-component series system under a FRPW policy which offers free replacement of the failed component if any system failure happens within the warranty period w. Recently, *Bai* and *Pham* [7.39] obtained the first two moments of a renewable FSW policy for series, parallel, series–parallel and parallel–series systems. The derivation of the first two moments of the DWC of nonrenewable FRPW and PRW policies for minimally repaired series systems can be found in [7.58].

It is possible to use a Markovian model to analyze warranty cost for complex systems. *Balachandran* et al. [7.59] dealt with the problem of determining warranty service cost of a three-component system using the Markovian approach. A similar discussion can be seen in [7.55] and the references therein. Although this approach is a powerful tool in the literature of reliability, queuing systems, and supply-chain management, there are some limitations in the applications of warranty. First of all, it is difficult to determine the appropriate state space and the corresponding transition matrix for the applications in warranty. Secondly, most Markovian models only provide the analysis of measures in the steady states by assuming infinite horizon. In other words, the statistical behavior of those measures in finite horizon (short-run) is either too difficult to obtain or not of practical interest. However, in warranty study, it is crucial to understand the finite-horizon statistical behavior of warranty cost. Thirdly, warranty claim data as well as reliability data are scarce and costly. Markovian models usually require more data since they contain more parameters than ordinary probability models that could be applied to warranty cost study.

7.2.4 Assessing Warranty Benefits

As mentioned in the introduction, warranty is increasingly used as a promotional device for marketing purposes. Consequently, it is necessary to predict and assess quantitatively the benefit that a manufacturer might generate from a specific warranty [7.35, p. 189]. For promotional warranties, such benefit is usually realized through the demand side. Manufacturers generally expect that the increase in profit as a result of the increase in sale, which is boosted by warranty, should cover the future warranty cost.

A simple way to quantify the benefit is to model it as a function of the parameter(s) of a warranty policy, for example, w, the warranty period. A linear form and a quadratic form of w were employed by *Thomas* [7.35, 60] for this purpose. As he acknowledged, both forms were not well-founded and shared the problem of oversimplification [7.35, p. 193]. Another approach is to estimate the demand function empirically. *Menezes* and *Currim* [7.61] posited a general demand function where the quantity sold by a firm offering a warranty with period w is a function of its price, warranty length, advertising, distribution, quality, product feature, and the corresponding values for the firm's competitor. Based on the data from *Ward's Automotive Yearbook, Consumer Reports, Advertising Age, Leading National Advertisers*, and other sources during the period 1981–1987, they obtained the price elasticity and the warranty elasticity, which enabled them to obtain the optimal warranty length through maximizing the present value of cumulative future profit over a finite planning horizon. One of the limitations of this approach, as pointed out by the authors, is that it requires the support of historical sales data. As a result, it cannot be applied to new products or existing products without such historical data [7.61, p. 188].

A related problem of the demand side of warranty is the modeling of sales over time. Mahajan et al. presented several variant diffusion models that may be appropriate for consumer durables [7.62]. *Ja* et al. obtained the first two moments of warranty cost in a product life cycle by assuming a nonhomogeneous Poisson sale process [7.21]. It seems that such models do not allow the interaction between warranty and sales, therefore, they may not be used in estimating warranty benefit.

There is some research (*Emons* [7.15], *Lutz* and *Padmanabhan* [7.16], and *Padmanabhan* and *Rao* [7.17], etc.) on the demand side of warranty concerning moral hazard, advertising, consumers satisfaction, and so forth. However, compared to the vast warranty literature on estimating total warranty cost, the study on the demand side of warranty is far behind. Hopefully we will see more studies on this aspect in the future.

7.2.5 On the Optimal Warranty Policy

One of the most important objectives of warranty study is to assist warranty management. In particular, in the design phase of a warranty program, there are often a set of warranties that might be appropriate for a specific type of products. The problem faced by warranty managers therefore is how to determine the optimal warranty policy.

An early attempt to address the warranty design problem is based on the concept of life-cycle costing (*Blischke* [7.63], *Mamer* [7.64]). It is assumed that a consumer requires the product over a certain time period or life cycle from the same producer repeatedly upon each product failure no matter whether under warranty or not. Under this idealized producer–consumer relationship, the producer's life-cycle profit and the consumer's life-cycle cost can be calculated. Consequently, a consumer indifference price may be determined by comparing consumer's life-cycle costs with or without warranty. Similarly, the producer's indifference price may be calculated based on the comparison of the life-cycle profits with or without warranty.

An alternative approach is to set up an optimization problem to determine the optimal price and warranty length combination jointly through a game-theoretic perspective. In general, two parties, a warranty issuer and a representative consumer, participate in the game. The latter acts as a follower who responses rationally to each potential warranty offer by maximizing his/her utility. The former, as a leader, makes the decision on the optimal warranty strategy, which maximizes the expected profit, based on the anticipated rational response by the consumer. *Singpurwalla* and *Wilson* [7.5] studied two-attribute warranties through this approach. Some others references are *Chun* and *Tang* [7.65], *DeCroix* [7.13], *Glickman* and *Berger* [7.66], *Ritchken* [7.67], *Thomas* [7.60] and the references therein. In the context of production planning and marketing, *Mitra* and *Patankar* [7.68] presented a multi-criteria model that could be used in warranty design.

Now, we present a general formulation of the warranty design problem with some discussion, which may raise more interest among researchers and practitioners for further study.

Let $\Psi = \{\psi_1, \psi_2, \cdots, \psi_n\}$ represent the set of appropriate warranty policies for a given type of products. Policy ψ_i may contain more than one parameter. Denote by $\boldsymbol{w}_i$ the set of warranty parameters for ψ_i; then we can represent ψ_i by $\psi(\boldsymbol{w}_i)$ or $\boldsymbol{w}_i$. If $\boldsymbol{w}_i$ contains only one parameter, say, w_i, the warranty period, then $\boldsymbol{w}_i = \{w_i\}$. Denote by $p(\boldsymbol{w}_i)$ the selling price under the policy ψ_i, and by $C_j(\boldsymbol{w}_i)$ the random warranty cost for the j-th product sold under the policy ψ_i. Let p_0 be the production cost per unit (not including the warranty cost), then the optimal warranty policy $\psi(\boldsymbol{w}^*)$ may be obtained by solving

$$\begin{aligned}
&\max_{\{\boldsymbol{w}_i, \forall i, i=1,2,\cdots,n\}} \mathrm{E}\{\mathrm{U}[\pi(\boldsymbol{w}_i)]\} \\
&\text{s.t.} \quad w_i^l \le w_i \le w_i^u, \forall i, i = 1, 2, \cdots, n \\
&\mathrm{P}\left[\sum_{j=1}^{d(\boldsymbol{w}_i)} C_j(\boldsymbol{w}_i) \ge R_0\right] \le \alpha, \forall i, i = 1, 2, \cdots, n\,,
\end{aligned}$$

where $U(\cdot)$ is the monetary utility function that reflects the risk attitude of the manufacturer. It is a linear function if the manufacturer is risk-neutral and a concave function in the case of a risk-averse manufacturer; $\pi(\boldsymbol{w}_i) = \sum_{j=1}^{d(\boldsymbol{w}_i)} [p(\boldsymbol{w}_i) - p_0 - C_j(\boldsymbol{w}_i)]$; w_i^l, w_i^u are some lower and upper bounds of w_i; $d(\boldsymbol{w}_i)$ represents the demand function for $\psi(\boldsymbol{w}_i)$; R_0 is the predetermined warranty budget level; and α is the risk-tolerance level of the manufacturer with regard to R_0.

One should note that the second set of constraints is actually related to *value at risk* (VaR), a concept widely used in risk management, which indicates the maximum percentage value of an asset that could be lost during a fixed period within a certain confidence level [7.69]. It is reasonable to assume that manufacturers want to control VaR such that the probability that the total warranty cost is over the budget is within the accepted level α.

Solving the optimization problem might be a challenge. First of all, it is difficult to determine the demand function $d(\boldsymbol{w}_i)$, although it is possible to estimate it through marketing surveys or historical data. Secondly, it is required that warranty managers have complete knowledge of the selling price $p(\boldsymbol{w}_i)$. This requires a pricing strategy in the design phase of warranty. It should be noted that we could have considered $p(\boldsymbol{w}_i)$ as one of the decision variables, but this makes the problem more complicated. Besides, it is not rare in practice that

the price is simply set by adding a fixed margin over the estimated production cost with warranty. Thirdly, it is required that the probability distribution of warranty cost should be known. Little research has been done with regard to this issue except *Polatoglu* and *Sahin* [7.27] and *Sahin* and *Polatoglu* [7.70]. In general, numerical methods are required for this purpose. Fourthly, the problem is formulated as a nonlinear optimization problem with some constraints, which may be solved by nonlinear optimization software such as GAMS. However, in general there is no guarantee of the existence of a global optimal solution.

7.3 Concluding Remarks

A warranty problem, by its nature, is a multi-disciplinary research topic. Many researchers ranging from the industry engineer, economist, statistician, to marketing researchers have contributed greatly to warranty literature. In this chapter, we present an overview of warranty policies, focusing on the cost and benefit analysis from warranty issuers' perspective. Although we have successfully addressed several problems in this area, there are still a lot of opportunities for future research, a few of which are listed below:

- To advance warranty optimization models and perform empirical study based on the new developed models.
- To develop and apply efficient algorithms to solve warranty optimization problems.
- To propose and analyze new warranty policies appropriate for complex systems.
- To Study the distribution and the moments of discounted warranty cost for various policies.
- Warranty cost modeling for systems with more complex structures, including standby systems, bridge systems and network systems, etc.
- Develop warranty models considering failure dependency between components due to environmental impact.

References

7.1 W. R. Blischke, D. N. P. Murthy: *Warranty Cost Analysis* (Marcel Dekker, New York 1994)
7.2 W. R. Blischke, D. N. P. Murthy: Product warranty management–I: a taxonomy for warranty policies, Eur. J. Oper. Res. **62**, 127–148 (1993)
7.3 W. R. Blischke, D. N. P. Murthy (Eds.): *Product Warranty Handbook* (Marcel Dekker, New York 1996)
7.4 E. P. McGuire: *Industrial Product Warranties: Policies and Practices* (The Conference Board, New York 1980)
7.5 N. D. Singpurwalla, S. Wilson: The warranty problem: Its statistical, game theoretic aspects, SIAM Rev. **35**, 17–42 (1993)
7.6 J. Chen, D. D. Yao, S. Zheng: Quality control for products supplied with warranty, Oper. Res. **46**, 107–115 (1988)
7.7 I. Djamaludin, D. N. P. Murthy: Quality control through lot sizing for items sold with warranty, Int. J. Prod. Econ. **33**, 97–107 (1994)
7.8 G. G. Hegde, P. Kubat: Diagnosic design: A product support strategy, Eur. J. Oper. Res. **38**, 35–43 (1989)
7.9 Jie Mi: Warranty, burn-in, Naval Res. Logist. **44**, 199–210 (1996)
7.10 D. N. P. Murthy, A. Z. M. O. Hussain: Warranty, optimal redundancy design, Eng. Optim. **23**, 301–314 (1993)
7.11 D. G. Nguyen, D. N. P. Murthy: Optimal reliability allocation for products sold under warranty, Eng. Optim. **13**, 35–45 (1988)
7.12 I. Sahin: Conformance quality, replacement costs under warranty, Prod. Oper. Man. **2**, 242–261 (1993)
7.13 G. A. DeCroix: Optimal warranties, reliabilities, prices for durable goods in an oligopoly, Eur. J. Oper. Res. **112**, 554–569 (1999)
7.14 W. Emons: Warranties, moral hazard, the lemons problem, J. Econ. Theory **46**, 16–33 (1988)
7.15 W. Emons: On the limitation of warranty duration, J. Ind. Econ. **37**, 287–301 (1989)
7.16 M. A. Lutz, V. Padmanabhan: Warranties, extended warranties, product quality, Int. J. Ind. Organ. **16**, 463–493 (1998)
7.17 V. Padmanabhan, R. C. Rao: Warranty policy, extended service contracts: theory, an application to automobiles, Market. Sci **12**, 97–117 (1993)
7.18 D. N. P. Murthy, E. Asgharizadeh: Optimal decision making in a maintenance service operation, Eur. J. Oper. Res. **116**, 259–273 (1999)
7.19 E. W. Frees: Warranty analysis, renewal function estimation, Naval Res. Logist. Quart. **33**, 361–372 (1986)
7.20 E. W. Frees: Estimating the cost of a warranty, J. Bus. Econ. Stat. **6**, 79–86 (1988)

7.21 S. Ja, V. Kulkarni, A. Mitra, G. Patankar: Warranty reserves for non-stationary sales processes, Naval Res. Logist. **49**, 499–513 (2002)

7.22 J. D. Kalbfleisch, J. F. Lawless, J. A. Robinson: Methods for the analysis, prediction of warranty claims, Technometrics **33**, 273–285 (1991)

7.23 E. P. C. Kao, M. S. Smith: Computational approximations of renewal process relating to a warranty problem: the case of phase-type lifetimes, Eur. J. Oper. Res. **90**, 156–170 (1996)

7.24 E. P. C. Kao, M. S. Smith: On excess, current, total life distributions of phase-type renewal processes, Naval Res. Logist. **39**, 789–799 (1992)

7.25 U. Menzefricke: On the variance of total warranty claims, Comm. Statist. Theory Methods **21**, 779–790 (1992)

7.26 J. G. Patankar, G. H. Worm: Prediction intervals for warranty reserves, cash flows, Man. Sci. **27**, 237–241 (1981)

7.27 H. Polatoglu, I. Sahin: Probability distribution of cost, revenue, profit over a warranty cycle, Eur. J. Oper. Res. **108**, 170–183 (1998)

7.28 Y. H. Chun, K. Tang: Cost analysis of two-attribute warranty policies based on the product usage rate, IEEE Trans. Eng. Man. **46**, 201–209 (1999)

7.29 Y. Lam, P. K. W. Lam: An extended warranty policy with options open to consumers, Eur. J. Oper. Res. **131**, 514–529 (2001)

7.30 C. Wang, S. Sheu: Optimal lot sizing for products sold under free-repair warranty, Eur. J. Oper. Res. **164**, 367–377 (2005)

7.31 R. H. Yeh, W. T. Ho, S. T. Tseng: Optimal production run length for products sold with warranty, Eur. J. Oper. Res. **120**, 575–582 (2000)

7.32 R. H. Yeh, H. Lo: Optimal preventive–maintenance warranty policy for repairable products, Eur. J. Oper. Res. **134**, 59–69 (2001)

7.33 D. N. P. Murthy: Product warranty management–III: A review of mathematical models, Eur. J. Oper. Res. **62**, 1–34 (1992)

7.34 D. N. P. Murthy, I. Djamaludin: New product warranty: a literature review, Int. J. Prod. Econ. **79**, 231–260 (2002)

7.35 M. U. Thomas, S. S. Rao: Warranty economic decision models: A summary, some suggested directions for future research, Oper. Res. **47**, 807–820 (1999)

7.36 R. E. Barlow, F. Proschan: *Mathematical Theory of Reliability* (Wiley, New York 1965)

7.37 H. Pham, H. Wang: Imperfect maintenance, Eur. J. Oper. Res. **94**, 425–438 (1996)

7.38 H. Wang: Reliability and maintenance modeling for systems with imperfect maintenance and dependence. Ph.D. Thesis (Rutgers University, Piscataway 1997) (unpublished)

7.39 J. Bai, H. Pham: Cost analysis on renewable full-service warranties for multi-component systems, Eur. J. Oper. Res. **168**, 492–508 (2006)

7.40 Y. H. Chun: Optimal number of periodic preventive maintenance operations under warranty, Reliab. Eng. Sys. Saf. **37**, 223–225 (1992)

7.41 N. Jack, J. S. Dagpunar: An optimal imperfect maintenance policy over a warranty period, Microelectron. Reliab. **34**, 529–534 (1994)

7.42 J. S. Dagpunar, N. Jack: Optimal repair-cost limit for a consumer following expiry of a warranty, IMA J. Math. Appl. Bus. Ind. **4**, 155–161 (1992)

7.43 I. Sahin, H. Polatoglu: Maintenance strategies following the expiration of warranty, IEEE Trans. Reliab. **45**, 221–228 (1996)

7.44 H. Wang: A survey of maintenance policies of deteriorating systems, Eur. J. Oper. Res. **139**, 469–489 (2002)

7.45 J. Bai, H. Pham: RLRF warranty policies with imperfect repair: A censored quasirenewal process approach, working paper, Department of Industrial and Systems Engineering, Rutgers University (2003)

7.46 D. G. Nguyen, D. N. P. Murthy: Optimal replacement–repair strategy for servicing products sold under warranty, Eur. J. Oper. Res. **39**, 206–212 (1989)

7.47 D. N. P. Murthy, B. P. Iskandar, R. J. Wilson: Two dimensional failure-free warranty policies: two-dimensional point process models, Oper. Res. **43**, 356–366 (1995)

7.48 H. Moskowitz, Y. H Chun: A Poisson regression model for two-attribute warranty policies, Naval Res. Logist. **41**, 355–376 (1994)

7.49 H. G. Kim, B. M. Rao: Expected warranty cost of two-attribute free-replacement warranties based on a bivariate exponential distribution, Comput. Ind. Eng. **38**, 425–434 (2000)

7.50 E. A. Elsayed: *Reliability Engineering* (Addison Wesley Longman, Reading 1996)

7.51 V. Lee Hill, C. W. Beall, W. R. Blischke: A simulation model for warranty analysis, Int. J. Prod. Econ. **16**, 463–491 (1998)

7.52 D. M. Kreps: *A Course in Microeconomic Theory* (Princeton Univ. Press, Princeton 1990)

7.53 H. Markowitz: *Portfolio Selection* (Yale Univ. Press, Yale 1959)

7.54 P. H. Ritchken, C. S. Tapiero: Warranty design under buyer, seller risk aversion, Naval Res. Logist. Quart. **33**, 657–671 (1986)

7.55 S. Chukova, B. Dimitrov: Warranty analysis for complex systems. In: *Product Warranty Handbook*, ed. by W. R. Blischke, D. N. P. Murthy (Marcel Dekker, New York 1996) pp. 543–584

7.56 P. H. Ritchken: Optimal replacement policies for irreparable warranted item, IEEE Trans. Reliab. **35**, 621–624 (1986)

7.57 A. Z. M. O. Hussain, D. N. P. Murthy: Warranty, redundancy design with uncertain quality, IEEE Trans. **30**, 1191–1199 (1998)

7.58 J. Bai, H. Pham: Discounted warranty cost for minimally repaired series systems., IEEE Trans. Reliab. **53**(1), 37–42 (2004)

7.59 K. R. Balachandran, R. A. Maschmeyer, J. L. Livingstone: Product warranty period: A Markovian approach to estimation, analysis of repair, replacement costs, Acc. Rev. **1**, 115–124 (1981)

7.60 M. U. Thomas: Optimum warranty policies for nonreparable items, IEEE Trans. Reliab. **32**, 283–288 (1983)

7.61 M. Menezes: An approach for determination of warranty length, Int. J. Res. Market. **9**, 177–195 (1992)

7.62 V. Mahajan, E. Muller, Y. Wind: *New-product diffusion models* (Kluwer Academic, Dordrecht 2000)

7.63 W. R. Blischke, E. M. Scheuer: Calculating the cost of warranty policies as a function of estimated life distributions, Naval Res. Logist. Quart. **28**, 193–205 (1975)

7.64 J. W. Mamer: Discounted, per unit costs of product warranty, Man. Sci. **33**, 916–930 (1987)

7.65 Y. H. Chun, K. Tang: Determining the optimal warranty price based on the producer's, customers' risk preferences, Eur. J. Oper. Res. **85**, 97–110 (1995)

7.66 T. S. Glickman, P. D. Berger: Optimal price, protection period decisions for a product under warranty, Man. Sci. **22**, 1381–1390 (1976)

7.67 P. H. Ritchken: Warranty policies for non-repairable items under risk aversion, IEEE Trans. Reliab. **34**, 147–150 (1985)

7.68 A. Mitra, J. G. Patankar: An integrated multicriteria model for warranty cost estimation, production, IEEE Trans. Eng. Man. **40**, 300–311 (1993)

7.69 P. Jorion: *Value-at-Risk: The New Benchmark for Managing Financial Risk* (McGraw-Hill, New York 2000)

7.70 I. Sahin, H. Polatoglu: Distributions of manufacturer's, user's replacement costs under warranty, Naval Res. Logist. **42**, 1233–1250 (1995)

8. Stationary Marked Point Processes

Many areas of engineering and statistics involve the study of a sequence of random events, described by points occurring over time (or space), together with a *mark* for each such point that contains some further information about it (type, class, etc.). Examples include image analysis, stochastic geometry, telecommunications, credit or insurance risk, discrete-event simulation, empirical processes, and general queueing theory. In telecommunications, for example, the events might be the arrival times of requests for bandwidth usage, and the marks the bandwidth capacity requested. In a mobile phone context, the points could represent the locations (at some given time) of all mobile phones, and the marks 1 or 0 as to whether the phone is in use or not. Such a stochastic sequence is called a *random marked point process*, an MPP for short. In a stationary stochastic setting (e.g., if we have moved our origin far away in time or space, so that moving further would not change the distribution of what we see) there are two versions of an MPP of interest depending on how we choose our origin: point-stationary and time-stationary (space-stationary). The first randomly chooses an event point as the origin, whereas the second randomly chooses a time (or space) point as the origin. Fundamental mathematical relationships exists between these two versions allowing for nice applications and computations. In what follows, we present this basic theory with emphasis on one-dimensional processes over time, but also include some recent results for d-dimensional Euclidean space, $\mathbb{R}^d$.

This chapter will primarily deal with marked point processes with points on the real line (time). Spatial point processes with points in $\mathbb{R}^d$ will be touched upon in the final section; some of the deepest results in multiple dimensions have only come about recently.

Topics covered include point- and time-stationarity, inversion formulas, the Palm distribution, Campbell's formula, MPPs jointly with a stochastic process, the rate conservation law, conditional intensities, and ergodicity.

8.1 Basic Notation and Terminology

Here the basic framework is presented for MPPs on the real line, with the points distributed over time.

8.1.1 The Sample Space as a Sequence Space

A widely used class of MPPs has events corresponding to points in time,

$$0 \le t_0 < t_1 < t_2 < \cdots , \ \lim_{n\to\infty} t_n = \infty . \tag{8.1}$$

An MPP is then defined as a stochastic sequence; a sequence of random variable (RVs),

$$\Psi = \{(t_n, k_n) : n \ge 0\} ,$$

where the marks k_n take values in a general space $\mathbb{K}$, the mark space, which is assumed to be a complete separable metric space, where the sample-paths of Ψ satisfy (8.1). (It helps to imagine that the arrivals correspond to *customers* arriving to some fixed location over time, each one bringing with them an object called their mark: the n-th customer arrives at time t_n and brings mark k_n.) Alternatively, with $T_n \stackrel{\text{def}}{=} t_{n+1} - t_n, \ n \ge 0$ denoting the n-th *interevent* (interarrival) time, Ψ can equivalently be defined by its *interevent time representation* $\{t_0, \{(T_n, k_n) : n \ge 0\}\}$.

Letting $\mathbb{R}_+$ and $\mathbb{Z}_+$ denote the non-negative real numbers and non-negative integers respectively, $\mathbb{S} = (\mathbb{R}_+ \times \mathbb{K})^{\mathbb{Z}_+}$ denotes sequence space, endowed with the product topology and corresponding Borel σ-field. $\boldsymbol{s} = \{(y_n, k_n) : n \in \mathbb{Z}_+\} \in \mathbb{S}$ denotes a sequence.

$\mathbb{M} \stackrel{\text{def}}{=} \{\boldsymbol{s} \in \mathbb{S} : \boldsymbol{s}$ satisfies (8.1)$\}$, and is the space of marked point processes with mark space $\mathbb{K}$, that is, the *MPP space*. Elements of $\mathbb{M}$ are denoted by $\psi = \{(t_n, k_n)\} \in \mathbb{M}$; they are the sample paths of an MPP $\Psi : \Omega \to \mathbb{M}$, formally a mapping from a probability space Ω into $\mathbb{M}$ with some underlying probability P. [It is standard to suppress the dependency of the random elements on $\omega \in \Omega$; e.g., $t_n(\omega), k_n(\omega), \ \Psi(\omega)$.] When $\Omega = \mathbb{M}$, this is called the *canonical* representation of Ψ. The sequence of points themselves, without marks, $\{t_n\}$, is called a *point process*.

The probability distribution of Ψ is denoted by $\boldsymbol{P} \stackrel{\text{def}}{=} P(\Psi \in \cdot)$; it is a distribution on the Borel sets $\mathcal{E} \subset \mathbb{M}$; $\boldsymbol{P}(\mathcal{E}) = P(\Psi \in \mathcal{E})$.

Two MPPs Ψ_1 and Ψ_2 are said to have the same distribution if $P(\Psi_1 \in \mathcal{E}) = P(\Psi_2 \in \mathcal{E})$ for all Borel sets $\mathcal{E} \subset \mathbb{M}$; equivalently *all finite-dimensional distributions of the two sequences are identical*, e.g., they agree for all Borel sets of the form

$$\mathcal{E} = \{\psi \in \mathbb{M} : t_{n_0} \le s_0, k_{n_0} \in K_0, \ldots , \\ t_{n_l} \le s_l, k_{n_l} \in K_l\} ,$$

where $0 \le n_0 < \cdots < n_l, l \ge 0, s_i \ge 0, K_i \subset \mathbb{K}, 0 \le i \le l$.

The assumption (8.1) of strict monotonicity, $t_n < t_{n+1}, \ n \ge 0$, can be relaxed to $t_n \le t_{n+1}, \ n \ge 0$, to accommodate *batch* arrivals, such as busloads or other groups that arrive together, but if the inequalities are strict, then the MPP is called a *simple* MPP.

8.1.2 Two-sided MPPs

With $\mathbb{Z}$ denoting all integers, a *two-sided* MPP, $\Psi = \{(t_n, k_n) : n \in \mathbb{Z}\}$, has points defined on all of the real line $\mathbb{R}$ thus allowing for arrivals since the infinite past;

$$\cdots t_{-2} < t_{-1} < t_0 \le 0 < t_1 < t_2 < \cdots . \tag{8.2}$$

(In this case, by convention, $t_0 \le 0$.)

8.1.3 Counting Processes

For an MPP $\psi \in \mathbb{M}$, let $N(t) = \sum_j I\{t_j \in (0, t]\}$ denote the number of points that occur in the time interval $(0, t], \ t > 0$. ($I\{B\}$ denotes the indicator function for the event B.) $\{N(t) : t \ge 0\}$ is called the *counting process*. By convention $N(0) \stackrel{\text{def}}{=} 0$. For $0 \le s \le t$, $N(s, t] \stackrel{\text{def}}{=} N(t) - N(s)$, the number of points in $(s, t]$.

In a two-sided framework, counting processes can be extended by defining $N(-t) = \sum_j I\{t_j \in (-t, 0]\}$, the number of points in $(-t, 0], \ t \ge 0$. In this case

$$t_j = \begin{cases} \inf\{t > 0 : N(t) \ge j\}, & j \ge 1 ; \\ -\inf\{t > 0 : N(-t) \ge j + 1\}, & j \le 0 , \end{cases}$$

and, for $t > 0$, $N(t) = \max\{j \ge 1 : t_j \le t\}$; $t_{N(t)}$ is thus the last point before or at time t, and $t_{N(t)+1}$ is the first point strictly after time t; $t_{N(t)} \le t < t_{N(t)+1}$. $T_{N(t)} = t_{N(t)+1} - t_{N(t)}$ is the interarrival time that covers t. Note that $\{t_j \le t\} = \{N(t) \ge j\}, \ j \ge 1$: an obvious but useful identity. For example, in a stochastic setting it yields $P(N(t) = 0) = P(t_1 > t)$. [In the one-sided case, $P(N(t) = 0) = P(t_0 > t)$.]

For a fixed mark set $K \subset \mathbb{K}$, let $N_K(t) = \sum_j I\{t_j \in (0, t], \ k_j \in K\}$, the counting process of points restricted to the mark set K. The MPP corresponding to $\{N_K(t)\}$ is sometimes referred to as a *thinning* of ψ by the mark set K.

Counting processes uniquely determine the MPP, and can be extended to measures, as will be presented in Sect. 8.1.5.

8.1.4 Forward and Backward Recurrence Times

The *forward recurrence time* is defined by

$$A(t) \stackrel{\text{def}}{=} t_{N(t)+1} - t = \begin{cases} t_0 - t, & \text{if } 0 \le t < t_0 \,; \\ t_{n+1} - t, & \text{if } t_n \le t < t_{n+1},\ n \in \mathbb{Z}_+ \,. \end{cases}$$

It denotes the time until the next event strictly after time t and is also called the *excess* at time t. At an arrival time t_n, $A(t_n-) = 0$ and $A(t_n) = A(t_n+) = T_n$, then it decreases down to zero linearly with rate one, making its next jump at time t_{n+1} and so on.

Similarly we can define the *backward recurrence time*

$$B(t) \stackrel{\text{def}}{=} t - t_{N(t)} = \begin{cases} t, & \text{if } 0 \le t < t_0 \,; \\ t - t_n, & \text{if } t_n \le t < t_{n+1},\ n \in \mathbb{Z}_+ \,, \end{cases}$$

which denotes the time since the last event prior to or at time t. In particular, $B(t) \le t$ and $B(0) = 0$. $B(t)$ is also called the *age* at time t. At an arrival time t_{n+1}, $B(t_{n+1}-) = T_n$ and $B(t_{n+1}+) = 0$ and then increases to T_{n+1} linearly with rate one. The sample paths of A and B are mirror images of each other.

In a two-sided framework, $A(t) = t_{n+1} - t$ and $B(t) = t - t_n$, if $t_n \le t < t_{n+1}$, $n \in \mathbb{Z}$; $B(t)$ is no longer bounded by t, $B(0) = |t_0|$ and $A(0) = t_1$ [recall (8.2)].

$S(t) = B(t) + A(t) = t_{N(t)+1} - t_{N(t)} = T_{N(t)}$ is called the *spread* or *total lifetime* at time t; $S(t) = T_n$ if $t_n \le t < t_{n+1}$, and is therefore piecewise constant. In a two-sided framework, $S(0) = |t_0| + t_1$.

In the context of consecutively replaced light bulbs at times t_n with lifetimes $\{T_n\}$, $A(t)$ denotes the remaining lifetime of the bulb in progress at time t, while $B(t)$ denotes its age. $S(t)$ denotes the total lifetime of the bulb in progress.

8.1.5 MPPs as Random Measures: Campbell's Theorem

An MPP ψ can equivalently be viewed as a σ-finite $\mathbb{Z}_+$ valued measure

$$\psi = \sum_j \delta_{(t_j, k_j)} \,,$$

on (the Borel sets of) $\mathbb{R}\times\mathbb{K}$, where $\delta_{(t_j,k_j)}$ is the Dirac measure at (t_j, k_j). For $A \subset \mathbb{R}$ and $K \subset \mathbb{K}$, $\psi(A\times K) =$ the number of points that occur in the time set A with marks taking values that fall in K;

$$\psi(A\times K) = \sum_j I(t_j \in A,\ k_j \in K) \,.$$

$\psi(A\times\mathbb{K}) < \infty$ for all bounded sets A. If $g = g(t,k)$ is a real-valued measurable function on $\mathbb{R}\times\mathbb{K}$, then the integral $\psi(g)$ is given by

$$\psi(g) = \int g \,\mathrm{d}\psi = \int g(t,k)\psi(\mathrm{d}t,\,\mathrm{d}k) = \sum_j f(t_j, k_j) \,.$$

An MPP Ψ can thus be viewed as a random measure and ν denotes its *intensity measure* on $\mathbb{R}\times\mathbb{K}$, defined by $\nu(A\times K) = E[\Psi(A\times K)]$, the expected value; $\nu(\mathrm{d}t,\,\mathrm{d}k) = E[\Psi(\mathrm{d}t,\,\mathrm{d}k)]$. Starting first with simple functions of the form $g(t,k) = I\{t \in A, k \in K\}$ and then using standard approximation arguments leads to

Theorem 8.1 (Campbell's theorem)
For any non-negative measurable function $g = g(t,k)$,

$$E\big[\Psi(g)\big] = \int g \,\mathrm{d}\nu \,.$$

8.1.6 Stationary Versions

An MPP can be stationary in one of two ways, either with respect to point shifts or time shifts (but not both); the basics are presented here.

Define for each $s \ge 0$, the MPP ψ_s by

$$\psi_s = \{[t_n(s), k_n(s)] : n \in \mathbb{Z}_+\} \stackrel{\text{def}}{=} \{(t_{N(s)+n+1} - s,\, k_{N(s)+n+1});\, n \in \mathbb{Z}_+\} \,, \qquad (8.3)$$

the MPP obtained from ψ by shifting to s as the origin and relabeling the points accordingly. For $s \ge 0$ fixed, there is a unique $m \ge 0$ such that $t_m \le s < t_{m+1}$, in which case $t_0(s) = t_{m+1} - s$; $t_1(s) = t_{m+2} - s$; and $t_n(s) = t_{m+n+1} - s$ for $n \in \mathbb{Z}_+$. Similarly, the marks become $k_0(s) = k_{m+1}$; and $k_n(s) = k_{m+n+1}$ for $n \in \mathbb{Z}_+$.

When choosing $s = t_j$, a particular point, then ψ_s is denoted by $\psi_{(j)}$. In this case ψ is shifted to the point t_j so $\psi_{(j)}$ always has its initial point at the origin: $t_0(t_j) = 0$, $j \ge 0$.

The mappings from $\mathbb{M} \to \mathbb{M}$ taking ψ to ψ_s and ψ to $\psi_{(j)}$ are called *shift mappings*.

Applying these shifts to the sample paths of an MPP Ψ yields the shifted MPPs Ψ_s and $\Psi_{(j)}$. It is noteworthy

that, while Ψ_s is a deterministic shift of Ψ, $\Psi_{(j)}$ is a random shift because $t_j = t_j(\omega)$ depends upon the sample path.

In a two-sided framework, the shifts also include (and relabel) all points to the left of s, and s can be negative too.

Point Stationarity

Definition 8.1

Ψ is called a point-stationary MPP if $\Psi_{(j)}$ has the same distribution as Ψ for all $j \in \mathbb{Z}_+$. Equivalently its representation $\{t_0, \{(T_n, k_n) : n \in \mathbb{Z}_+\}\}$ has the properties that $P(t_0 = 0) = 1$ and $\{(T_n, k_n) : n \in \mathbb{Z}_+\}$ forms a stationary sequence of RVs.

If $\{(T_n, k_n) : n \in \mathbb{Z}_+\}$ is also ergodic, then Ψ is said to be a point-stationary and ergodic MPP.

For simplicity, we will always assume that a point-stationary MPP is ergodic.

In practical terms, ergodicity means that, for any measurable $f : \mathbb{M} \longrightarrow \mathbb{R}_+$,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} f(\Psi_{(j)}) = E(f(\Psi)) \,, \quad \text{with probability 1 (wp1)} \,. \tag{8.4}$$

(This is *Birkoff's ergodic theorem in its ergodic form*.) For example, if $f(\psi) = T_0$, then $f(\psi_{(j)}) = T_j$ and (8.4) yields the strong law of large numbers for the stationary ergodic sequence $\{T_n\}$; $\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} T_j = E(T_0)$, wp1. (The non-ergodic case is discussed in Sect. 8.8.)

Inherent in the definition of point-stationarity is the fact that there is a one-to-one correspondence between point-stationary point processes and stationary sequences of non-negative RVs; given any such stationary sequence $\{T_n\}$, $t_n \stackrel{def}{=} T_0 + \cdots + T_{n-1}$ (and $t_0 \stackrel{def}{=} 0$) defines a point-stationary point process.

When Ψ is point-stationary, we let T denote a generic interarrival time, define the *arrival rate* $\lambda = [E(T)]^{-1}$, and let $F(x) = P(T \le x)$, $x \ge 0$ denote the stationary interarrival time distribution with $\bar{F}(x) = 1 - F(x)$ being its tail. As in the classic *elementary renewal theorem*, it holds that $N(t)/t \to \lambda$ as $t \to \infty$, wp1.

From Kolmogorov's extension theorem in probability theory, a stationary sequence can be extended to be two-sided, $\{(T_n, k_n) : -\infty < n < \infty\}$, yielding a point-stationary MPP on all of $\boldsymbol{R}$:

$$\cdots t_{-2} < t_{-1} < t_0 = 0 < t_1 < t_2 < \cdots \,,$$

where $t_{-n} \stackrel{\text{def}}{=} -(T_{-1} + \cdots + T_{-n})$, $n \ge 1$.

Point-stationary MPPs arise naturally as limits (in distribution) of $\Psi_{(j)}$ as $j \to \infty$. In applications the limit can be taken in a *Cesàro* sense. Independently take a discrete RV J with a uniform distribution on $\{1, \ldots, n\}$, and define an MPP Ψ^0 by defining its distribution as

$$\boldsymbol{P}^0(\cdot) = P(\Psi^0 \in \cdot) \stackrel{\text{def}}{=} \lim_{n\to\infty} P(\Psi_{(J)} \in \cdot) = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} P(\Psi_{(j)} \in \cdot) \,. \tag{8.5}$$

If the limit holds for all Borel sets of $\mathbb{M}$, then it can be shown that it holds uniformly over all Borel sets; known as *Cesàro total variation convergence*. Assuming the existence of such a limiting distribution $\boldsymbol{P}^0$, it is unique and is called the *point-stationary distribution* of Ψ (or of $\boldsymbol{P}$) and Ψ is said to be *point asymptotically stationary*. Any MPP $\Psi^0 = \{(t_n^0, k_n^0)\}$ distributed as $\boldsymbol{P}^0$ is called a *point-stationary version* of Ψ. Intuitively this is obtained from Ψ by randomly selecting a point t_j so far in the infinite future that shifting further to the next point t_{j+1} does not change the distribution; it is stationary with respect to such point shifts.

It is important to remember that a point-stationary MPP has (wp1) a point at the origin.

Time Stationarity

Definition 8.2

Ψ is called *time-stationary* if Ψ_s has the same distribution as Ψ, for all $s \ge 0$. In this case $P(t_0 > 0) = 1$ and $\{N_K(t) : t \ge 0\}$ has stationary increments for each mark set K.

When Ψ is time-stationary, the interevent time sequence $\{(T_n, k_n)\}$ will not be stationary in general; in particular, the distribution of T_j will generally be different for different choices of j. However, the stochastic process $\{A(t)\}$ is a stationary process.

Ergodicity is defined as requiring that the measure-preserving flow of shifts $\theta_s : \mathbb{M}$ to $\mathbb{M}$, $s \ge 0$, $\theta_s \psi = \psi_s$ be ergodic under the distribution of Ψ. (In the point-stationary case, ergodicity is equivalent to requiring that the measure-preserving shift map $\theta_{(1)} = \theta_{t_1}$ be ergodic.)

For simplicity, we will always assume that a time-stationary MPP is ergodic. In practical terms,

ergodicity means that, for any measurable $f : \mathbb{M} \longrightarrow \mathbb{R}_+$ (satisfying $\int_0^t f(\Psi_s)\,\mathrm{d}s < \infty,\ t \geq 0,$ wp1),

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t f(\Psi_s)\,\mathrm{d}s = E\big[f(\psi)\big],\ \text{wp1}\,. \tag{8.6}$$

When Ψ is time-stationary, the arrival rate is defined by $\lambda \stackrel{\text{def}}{=} E[N(1)]$ and it holds that $E[N(t)] = \lambda t,\ t \geq 0$. It also holds that $N(t)/t \to \lambda$ as $t \to \infty$, wp1.

Time-stationary MPPs can be extended to be two-sided

$$\cdots t_{-2} < t_{-1} < t_0 < 0 < t_1 < t_2 < \cdots\,, \tag{8.7}$$

where $P(t_0 < 0,\ t_1 > 0) = 1$. In this case $\{B(t)\}$ and $\{S(t)\}$ are stationary processes in which case $B(0) = |t_0|$ and $A(0) = t_1$ are identically distributed.

Time-stationary MPPs arise naturally as limits (in distribution) of Ψ_t as time $t \to \infty$. In applications the limit can be taken in a Cesàro sense: independently take a continuous RV, U, uniformly distributed over $(0, t)$, and define an MPP Ψ^* by defining its distribution as

$$\begin{aligned} \boldsymbol{P}^*(\cdot) = P(\Psi^* \in \cdot) &\stackrel{\text{def}}{=} \lim_{t\to\infty} P(\Psi_U \in \cdot) \\ &= \lim_{t\to\infty} \frac{1}{t} \int_0^t P(\Psi_s \in \cdot)\,\mathrm{d}s\,. \end{aligned} \tag{8.8}$$

If the limit holds for all Borel sets of $\boldsymbol{M}$, then it can be shown that it holds uniformly over all Borel sets; Cesàro total variation convergence. Assuming the existence of such a limiting distribution $\boldsymbol{P}^*$, it is unique and is called the *time-stationary distribution* of Ψ (or of $\boldsymbol{P}$) and Ψ is said to be *time asymptotically stationary*. Any MPP $\Psi^* = \{(t_n^*, k_n^*)\}$ distributed as $\boldsymbol{P}^*$ is called a *time-stationary version* of Ψ. Intuitively it is obtained from Ψ by randomly selecting a time t as the origin that is so far in the infinite future that shifting s time units further does not change the distribution; it is stationary with respect to such time shifts.

It is important to remember that a time-stationary MPP has (wp1) no point at the origin.

8.1.7 The Relationship Between Ψ, Ψ^0 and Ψ^*

Suppose that Ψ has a point-stationary version Ψ^0. What then is the time-stationary distribution of Ψ^0? Intuitively it should be the same as the time-stationary distribution of Ψ, and this turns out to be so:

Proposition 8.1

Ψ is point asymptotically stationary (defined as in (8.5)) with point-stationary (and ergodic) $\boldsymbol{P}^0$ under which $0 < \lambda < \infty$, if and only if Ψ is time asymptotically stationary (defined as in (8.8)) with time-stationary (and ergodic) $\boldsymbol{P}^*$ under which $0 < \lambda < \infty$. In this case $\boldsymbol{P}^*$ is the time-stationary distribution of $\boldsymbol{P}^0$, and $\boldsymbol{P}^0$ is the point-stationary distribution of $\boldsymbol{P}^*$. (All three of $\Psi,\ \Psi^0,\ \Psi^*$ share the same point- and time-stationary distributions.)

Because of the above proposition, Ψ is called *asymptotically stationary* if one (hence both) of $\boldsymbol{P}^0$, $\boldsymbol{P}^*$ exist with $0 < \lambda < \infty$.

Proposition 8.2

Suppose that Ψ is asymptotically stationary (and ergodic). Then the two definitions of the arrival rate λ coincide; $\lambda = E[N^*(1)] = [E(T^0)]^{-1}$. Moreover, the ergodic limits in (8.4) and (8.6) hold for all three MPPs, Ψ, Ψ^0, Ψ^* with the right-hand sides replaced by $E[f(\Psi^0)]$ and $E[f(\Psi^*)]$ respectively.

It turns out that, in fact, all three MPPs, Ψ, Ψ^0, Ψ^* *shift couple*, and that is the key to understanding the above two propositions ($\stackrel{\text{d}}{\sim}$ denotes "is distributed as"):

Proposition 8.3

If Ψ is asymptotically stationary, then there exist versions of $\Psi \stackrel{\text{d}}{\sim} \boldsymbol{P}$, $\Psi^0 \stackrel{\text{d}}{\sim} \boldsymbol{P}^0$, $\Psi^* \stackrel{\text{d}}{\sim} \boldsymbol{P}^*$ all on a common probability space together with three random times, S_1, S_2, S_2 such that $\Psi_{S_1} = \Psi^0_{S_2} = \Psi^*_{S_3}$. In other words, they share the same sample paths modulo some time shifts.

Given an asymptotically stationary MPP Ψ, the superscripts ^{0} and * are used to denote point- and time-stationary versions of all associated processes of Ψ. $\Psi^0 = \{(t_n^0, k_n^0)\}$, and $\Psi^* = \{(t_n^*, k_n^*)\}$ denote the two versions, and, for example, $\{(T_n^0, k_n^0)\}$ denotes the stationary sequence of interevent times and marks for Ψ^0, and T^0 denotes such a generic interevent time with F being its distribution; $F(x) = P(T^0 \leq x),\ x \geq 0$. $\{A^*(t)\}$ denotes the forward recurrence time process for Ψ^*, etc.

To illustrate the consequences of Proposition 8.8.2, suppose that $f(\psi) = t_0$. Then $f(\psi_s) = t_0(s) = A(s)$, forward recurrence time, and it holds that

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t A^*(s)\,\mathrm{d}s = E(t_0^*),\ \text{wp1}\,,$$

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t A(s)\,\mathrm{d}s = E(t_0^*),\ \text{wp1}\,,$$

$$\lim_{t\to\infty}\frac{1}{t}\int_0^t A^0(s)\,\mathrm{d}s = E(t_0^*),\ \text{wp1}\,.$$

8.1.8 Examples

Some simple examples are presented. In some of these examples, marks are left out for simplicity and to illustrate the ideas of stationarity better.

1. *Poisson process:* A (time-homogenous) Poisson process with rate λ has independent and identically distributed (iid) interarrival times $T_n,\ n \geq 0$ with an exponential distribution, $P(T \leq x) = 1 - \mathrm{e}^{-\lambda x},\ x \geq 0$. Its famous defining feature is that $\{N(t)\}$ has both stationary and independent increments, and that these increments have a Poisson distribution; $N(t)$ is Poisson-distributed with mean $E[N(t)] = \lambda t,\ t \geq 0$; $P[N(t) = n] = \left[\mathrm{e}^{-\lambda t}(\lambda t)^n\right]/n!,\ n \in \mathbb{Z}_+$.
   If we place t_0 at the origin, $t_0 = 0$, then the Poisson process is point-stationary, whereas if we (independently) choose t_0 distributed as exponential at rate λ, then the Poisson process becomes time-stationary. Thus, for a Poisson process, removing the point at the origin from Ψ^0 yields Ψ^*, while placing a point at the origin for Ψ^* yields Ψ^0. Observe that, by the memoryless property of the exponential distribution, $A(t)$ is distributed as exponential with rate λ for all $t \geq 0$.
   A two-sided time-stationary version is obtained as follows: Choose both $|t_0^*| = B^*(0)$ and $t_1^* = A^*(0)$ as iid with an exponential λ distribution. All interarrival times $T_n^*,\ -\infty < n < \infty$ are iid exponential at rate λ except for $T_0^* = t_1^* - t_0^* = B^*(0) + A^*(0) = S^*(0)$, the spread, which has an Erlang distribution (mean $2/\lambda$). That the distribution of T_0^* is different (larger) than T results from the *inspection paradox*: Randomly choosing the origin in time, we are more likely to land in a larger than usual interarrival time because larger intervals cover a larger proportion of the time line. $S^*(t)$ is distributed as Erlang (mean $2/\lambda$) for all $t \in \mathbb{R}$, by stationarity.
   The Poisson process is the unique simple point process with a counting process that possesses both stationary and independent increments.
2. *Renewal process*: Interarrival times $\{T_n : n \geq 0\}$, are iid with a general distribution $F(x) = P(T \leq x)$ and mean $\lambda^{-1} = E(T)$. If $t_0 = 0$ then the renewal process is point-stationary, and is called a non-delayed version of the renewal process. If instead, independently, $t_0 = A(0) > 0$ and has the *stationary excess* distribution, F_e, defined by

$$F_\mathrm{e}(x) = \lambda\int_0^x \bar{F}(y)\,\mathrm{d}y,\ x \geq 0\,, \tag{8.9}$$

   then the renewal process is time-stationary and $A^*(t)$ is distributed as F_e for all $t \geq 0$. (In the Poisson process case $F_\mathrm{e} = F$.) In general, when $t_0 > 0$ the renewal process is said to be *delayed*. For any renewal process (delayed or not) $\Psi_{(j)}$ always yields a point-stationary version Ψ^0 (for any $j \geq 0$), while Ψ_s always yields a delayed version with delay $t_0(s) = A(s)$. Only when this delay is distributed as F_e is the version time-stationary. As $s \to \infty$, the distribution of $A(s)$ converges (in a Cesàro total variation sense) to F_e; this explains why the distribution of Ψ_s converges (in a Cesàro total variation sense) to the time-stationary version we just described.
   A two-sided time-stationary version Ψ^* is obtained when $T_n^*,\ n \neq 0$ are iid distributed as F, and independently $[B^*(0), A^*(0)] = (|t_0^*|, t_1^*)$ has the joint distribution $P(|t_0^*| > x, t_1^* > y) = \bar{F}_\mathrm{e}(x+y),\ x \geq 0,\ y \geq 0$. Here, as for the Poisson process, $T_0^* = S^*(0)$ has, due to the inspection paradox, a distribution that is stochastically larger than F, $P(T_0^* > x) \geq P(T > x),\ x \geq 0$; this is called the *spread* distribution of F and has tail

$$P(T_0^* > x) = \lambda x\bar{F}(x) + \bar{F}_\mathrm{e}(x)\,; \tag{8.10}$$

   while $E(T_0^*) = E(T^2)/E(T)$. If F has a density $f(x)$, then the spread has a density $\lambda x f(x)$, which expresses the length biasing contained in the spread. F_e always has a density, $f_\mathrm{e}(x) = \frac{\mathrm{d}}{\mathrm{d}x}F_\mathrm{e}(x) = \lambda\bar{F}(x)$, whether or not F does.
3. *Compound renewal process*: Given the counting process $\{N(t)\}$ for a renewal process, and independently an iid sequence of RVs $\{X_n\}$ (called the jumps), with jump distribution $G(x) = P(X \leq x),\ x \in \mathbb{R}$, the process

$$X(t) = \sum_{j=1}^{N(t)} X_j,\ t \geq 0$$

   is called a compound renewal process with jump distribution G. A widely used special case is when the renewal process is a Poisson process, called a *compound Poisson process*.
   This can elegantly be modeled as the MPP $\Psi = \{(t_n, k_n)\}$, where $\{t_n\}$ are the points and $k_n = X_n$. Because it is assumed that $\{X_n\}$ is

independent of $\{t_n\}$, obtaining point and time-stationary versions merely amounts to joining in the iid marks to Example 2's renewal constructions: $k_n^0 = X_n = k_n^*$.

4. *Renewal process with marks depending on interarrival times*: Consider a two-sided renewal process and define the marks as $k_n = T_{n-1}$, the length of the preceding interarrival time. The interesting case is to construct a time-stationary version. This can be done by using the two-sided time-stationary version of the point process, $\{t_n^*\}$, from Example 2. Note that, for $n \neq 1$, the k_n^* are iid distributed as F, defined by $k_n^* = T_{n-1}^*$; only k_1^* is different (biased via the inspection paradox). $k_1^* = T_0^*$ and has the spread distribution.
5. *Cyclic deterministic*: Starting with interarrival time sequence $\{T_n\} = \{1, 2, 3, 1, 2, 3, 1, 2, 3 \dots\}$, Ψ^0 is given by defining $t_0^0 = 0$ and

$$\{T_n^0 : n \geq 0\} = \begin{cases} \{1, 2, 3, 1, 2, 3, \dots\}, \text{ wp} = 1/3\,; \\ \{2, 3, 1, 2, 3, 1, \dots\}, \text{ wp} = 1/3\,; \\ \{3, 1, 2, 3, 1, 2, \dots\}, \text{ wp} = 1/3\,. \end{cases} \tag{8.11}$$

(By randomly selecting a j and choosing t_j as the origin, we are equally likely to select a T_j with length 1, 2, or 3; $P(T^0 = i) = 1/3,\ i = 1, 2, 3.$) The two-sided extension is given by defining $t_0^0 = 0$ and

$$\{\dots, T_{-1}^0, T_0^0, T_1^0, \dots\} = \begin{cases} \{\dots, 3, 1, 2, \dots\}, \text{ wp} = 1/3; \\ \{\dots, 1, 2, 3, \dots\}, \text{ wp} = 1/3\,; \\ \{\dots, 2, 3, 1, \dots\}, \text{ wp} = 1/3\,. \end{cases}$$

A construction of Ψ^* is given as follows. Let U denote a random variable having a continuous uniform distribution over (0, 1). Then

$$\{t_0^*,\ \{T_n^* : n \geq 0\}\} = \begin{cases} U,\ \{2, 3, 1, 2, 3, 1 \dots\}, \text{ wp} = 1/6\,; \\ 2U,\ \{3, 1, 2, 3, 1, 2 \dots\}, \text{ wp} = 1/3\,; \\ 3U,\ \{1, 2, 3, 1, 2, 3 \dots\}, \text{ wp} = 1/2\,. \end{cases} \tag{8.12}$$

By randomly selecting a time s as the origin, we would land inside an interarrival time of length 1, 2, or 3 with probability 1/6, 1/3 and 1/2 respectively (they are proportions of time). Given that we land inside one of length i, $t_0(s)$ would be distributed as $iU,\ i = 1, 2, 3$ (e.g., uniform on $(0, i)$). Unlike $\{T_n^0 : n \geq 0\}$, $\{T_n^* : n \geq 0\}$ is not a stationary sequence because of the unequal probabilities in the mixture.

This illustrates the general fact that t_0^* has the stationary excess distribution $F_e(x)$ of the point-stationary distribution $F(x) = P(T^0 \leq x)$ [recall (8.9)]. In a two-sided extension, the distribution of $T_0^* = |t_0^*| + t_1^* = S^*(0)$ is the spread distribution of F; in this case $P(T_0^* = i) = i/6,\ i = 1, 2, 3$, and the joint distribution of $(|t_0^*|, t_1^*)$ is of the mixture form $(1-U, U), (2-2U, 2U), (3-3U, 3U)$ with probabilities $1/6,\ 1/3,\ 1/2$ respectively.

This example also illustrates the general fact that the time reversal of an MPP Ψ has a different distribution from Ψ; the sequence $\{T_n^0 : n \geq 0\}$ has a different distribution from that of the sequence $\{T_n^0 : n \leq 0\}$.

6. *Single-server queue*: t_n denotes the arrival time of the n-th customer, denoted by C_n, to a system (such as a bank with one clerk) that has one server behind which customers wait in queue (line) in a first-in-first-out manner (FIFO). Upon entering service, C_n spends an amount of time S_n with the server and then departs. D_n denotes the length of time that C_n waits in line before entering service and is called the *delay* of C_n in queue. Thus C_n enters service at time $t_n + D_n$ and departs at time $t_n + D_n + S_n$; $W_n = D_n + S_n$ is called the *sojourn time*. The total number of customers in the system at time t, is denoted by $L(t)$ and can be constructed from $\{W_n\}$;

$$L(t) = \sum_{j=1}^{N(t)} I(W_j > t - t_j), \tag{8.13}$$

because C_j is in the system at time t if $t_j \leq t$ and $W_j > t - t_j$.

Letting $\Psi = [(t_n, S_n)]$ yields an MPP, with marks $k_n = S_n$, called the *input* to the queueing model; from it the queueing processes of interest can be constructed. It is known that D_n satisfies the recursion $D_{n+1} = (D_n + S_n - T_n)_+,\ n \geq 0$, where $x_+ \stackrel{\text{def}}{=} \max(x, 0)$ denotes the positive part of x, and yet another MPP of interest is $\Psi = \{[t_n, (S_n, D_n)]\}$, where now $k_n = (S_n, D_n)$. Letting $\boldsymbol{D}_{(n)} = (D_{n+m} : m \geq 0)$, another important MPP with an infinite-dimensional mark space is $\Psi = \{[t_n, (S_n, \boldsymbol{D}_{(n)})]\}$, where $k_n = (S_n, \boldsymbol{D}_{(n)})$. The *workload* $V(t)$ is defined by $V(t) = D_n + S_n - (t - t_n),\ t \in [t_n, t_{n+1}),\ n \geq 0$, and $D_n = V(t_n-)$; it rep-

resents the sum of all remaining service times in the system at time t. It can also model the water level of a reservoir into which the amounts S_n are inserted at the times t_n while water is continuously drained out at rate 1.

A point-stationary version $\Psi^0 = \{[t_n^0, (S_n^0, D_n^0)]\}$ yields a stationary version of the delay sequence $\{D_n^0\}$ with stationary delay distribution $P(D \le x) = P(D_0^0 \le x)$, which is an important measure of congestion from the point of view of customers, as is its mean, $d \stackrel{\text{def}}{=} E(D)$, the average delay.

A time-stationary version $\Psi^* = \{[t_n^*, (S_n^*, D_n^*)]\}$ yields a time-stationary version of workload $\{V^*(t)\}$ and corresponding stationary distribution $P(V \le x) = P(V^*(0) \le x)$, which is an important measure of congestion from the point of view of the system, as is its mean, $E(V)$, is the average workload.

If the input MPP is asymptotically stationary (ergodic) with $0 < \lambda E(S^0) < 1$, then it is known that $\Psi = \{[t_n, (S_n, D_n)]\}$ is asymptotically stationary, e.g., the stationary versions and distributions for such things as delay and workload exist.

8.2 Inversion Formulas

Inversion formulas allow one to derive $\boldsymbol{P}^0$ from $\boldsymbol{P}^*$, and visa versa.

Theorem 8.2 (Inversion formulas)
Suppose that Ψ is asymptotically stationary (and ergodic) and $0 < \lambda < \infty$. Then

$$P(\Psi^* \in \cdot) = \lambda E\left[\int_0^{T_0^0} I(\Psi_s^0 \in \cdot)\,\mathrm{d}s\right], \tag{8.14}$$

$$P(\Psi^0 \in \cdot) = \lambda^{-1} E\left[\sum_{j=0}^{N^*(1)} I(\Psi_{(j)}^* \in \cdot)\right], \tag{8.15}$$

which, in functional form, become

$$E(f(\Psi^*)) = \lambda E\left[\int_0^{T_0^0} f(\Psi_s^0)\,\mathrm{d}s\right], \tag{8.16}$$

$$E(f(\Psi^0)) = \lambda^{-1} E\left[\sum_{j=0}^{N^*(1)} f(\Psi_{(j)}^*)\right]. \tag{8.17}$$

Recalling (8.6) and Proposition 8.8.2, it is apparent that (8.14) and (8.16) are generalizations (to a stationary ergodic setting) of the renewal reward theorem from renewal theory:

The time average equals the expected value over a cycle divided by the expected cycle length.

Here a cycle length is (by point stationarity) represented by any interarrival time, so the first one, $T_0^0 = t_1^0$, is chosen for simplicity. Equations (8.15) and (8.17) are the inverse [recalling (8.4)]:

The point average equals the expected value over a unit of time divided by the expected number of points during a unit of time.

Here a unit of time is (by time stationarity) represented by any such unit, so the first one, (0, 1], is chosen for simplicity.

8.2.1 Examples

The following examples illustrate how some well-known results that hold for renewal processes, involving the stationary excess distribution (8.9) and the inspection paradox and spread distribution (8.10) also hold in general. Throughout, assume that Ψ is asymptotically stationary (and ergodic).

1. *Stationary forward recurrence time*: $P(t_0^* \le x) = P[A^*(t) \le x] = F_e(x)$ where $F(x) = P(T^0 \le x)$. This is derived by applying (8.17) with $f(\psi) = I(t_0 > x)$: $f(\psi_s^0) = I[t_0^0(s) > x]$ and $t_0^0(s) = A^0(s) = t_1^0 - s, \quad s \in [0, t_1^0)$; $\int_0^{T_0^0} f(\Psi_s^0)\,\mathrm{d}s = \int_0^{T_0^0} I\{s < T_0^0 - x\}\,\mathrm{d}s = (T_0^0 - x)_+$. $\lambda E[(T_0^0 - x)_+] = \lambda \int_x^\infty \bar{F}(y)\mathrm{d}y = \bar{F}_e(x)$.
2. *Stationary backwards recurrence time*: $P[B(0)^* \le x] = F_e(x)$. Here, a two-sided framework must be assumed so that $B(0) = |t_0|$. Applying (8.17) with $f(\psi) = I[B(0) > x]$: $f(\Psi_s^0) = I[B^0(s) > x]$ where $B^0(s) = s, \ s \in [0, t_1^0)$; $\int_0^{T_0^0} f(\Psi_s^0)\,\mathrm{d}s = \int_0^{T_0^0} I(s > x)\,\mathrm{d}s = (T_0^0 - x)_+$. $\lambda E[(T_0^0 - x)_+] = \bar{F}_e(x)$.
3. *Stationary spread*: $P(T_0^* > x) = \lambda x \bar{F}(x) + \bar{F}_e(x)$. Here again, a two-sided framework must be assumed so that $S(0) = |t_0| + t_1$. Applying (8.17)

with $f(\psi) = I(T_0 > x)$: $f(\psi_s) = I[S(s) > x]$ and $S^0(s) = T_0^0, s \in [0, t_1^0)$; $\int_0^{T_0^0} f(\Psi_s^0)\,\mathrm{d}s = \int_0^{T_0^0} I(T_0^0 > x)\,\mathrm{d}s = T_0^0 I(T_0^0 > x)$. $\lambda E(T_0^0 I(T_0^0 > x)) = \lambda x \times \bar{F}(x) + \bar{F}_e(x)$ by carrying out the integration $E[T_0^0 I(T_0^0 > x)] = \int_0^\infty P(T_0^0 > y, T_0^0 > x)\,\mathrm{d}y$.

8.2.2 The Canonical Framework

In the canonical framework $\boldsymbol{E}$ denotes expectation under $\boldsymbol{P}$, $\boldsymbol{E^0}$ denotes expectation under $\boldsymbol{P}^0$ and $\boldsymbol{E}^*$ denotes expectation under $\boldsymbol{P}^*$ and $\Psi : \mathbb{M} \to \mathbb{M}$ is the identity map; $\Psi(\psi) = \psi$. This makes for some elegance and simplicity in notation. For example, the inversion formulas in functional form become

$$\boldsymbol{E}^*[f(\Psi)] = \lambda \boldsymbol{E^0}\left[\int_0^{T_0} f(\Psi_s)\,\mathrm{d}s\right],$$
$$\boldsymbol{E^0}[f(\Psi)] = \lambda^{-1}\boldsymbol{E}^*\left[\sum_{j=0}^{N(1)} f(\Psi_{(j)})\right]. \tag{8.18}$$

8.3 Campbell's Theorem for Stationary MPPs

Suppose that $\Psi = \Psi^*$ is time-stationary (and ergodic), with point-stationary version Ψ^0. From the inversion formula (8.15), $P(k^0 \in K) = \lambda^{-1} E\{\Psi^*[(0,1] \times K]\}$, yielding $E\{\Psi^*[(0,1] \times K]\} = \lambda P(k^0 \in K)$. This implies that the intensity measure from Campbell's theorem becomes $\nu(A \times K) = E[\Psi^*(A \times K)] = \lambda l(A) P(k^0 \in K)$, where $l(A)$ denotes Lebesgue measure {e.g., $E[\Psi^*(\mathrm{d}t \times \mathrm{d}k)] = \lambda\,\mathrm{d}t P(k^0 \in \mathrm{d}k)$}. This can be rewritten as $\nu(A \times K) = \lambda l(A) E[I(k_0^0 \in K)]$, in terms of the mark at the origin k_0^0 of Ψ^0. This yields

Theorem 8.3 [Campbell's theorem under stationarity (and ergodicity)]
For any non-negative measurable function $g = g(t, k)$,

$$E[\Psi^*(g)] = \lambda E\left[\int_{\mathbb{R}} g\big(t, k_0^0\big)\,\mathrm{d}t\right].$$

8.3.1 Little's Law

A classic application of Campbell's theorem in queueing theory is when $\Psi^* = [(t_n^*, W_n^*)]$ (two-sided) represents a time-stationary queueing model, where t_n^* is the arrival time of the n-th customer, and W_n^* their sojourn time. Using $g(t, w) = 0$, $t > 0$ and $g(t, w) = I(w > |t|)$, $t \le 0$ yields $\Psi^*(g) = \sum_{j \le 0} I(W_j^* > |t_j^*|) = L^*(0)$, denoting the time-stationary number of customers in the system at time $t = 0$ [recall (8.13)]. Campbell's theorem then yields $E[L^*(0)] = \lambda E(W^0)$, known as *Little's Law* or $L = \lambda w$.

8.3.2 The Palm–Khintchine Formula

Another application of interest for Campbell's theorem is the *Palm–Khintchine formula*: for all $n \ge 0$ and $t > 0$,

$$P[N^*(t) > n] = \lambda \int_0^t P[N^0(s) = n]\,\mathrm{d}s\,. \tag{8.19}$$

Proof: Since this result does not involve any marks, the marks can be replaced by new ones: define $k_j = \psi_{(j)}$. With these new marks Ψ^* remains stationary (and ergodic). For fixed $t > 0$ and $n \ge 0$, define $g(s, \psi) = I[0 \le s \le t,\ N(t-s) = n]$. Then

$$\Psi^*(g) = \sum_{j=1}^{N^*(t)} I\big(N^*(t_j, t] = n\big) = I[N^*(t) > n]\,,$$

where the last equality is obtained by observing that $N(t) > n$ if and only if there exists a j (unique) such that $t_j < t$ and there are exactly n more arrivals during $(t_j, t]$. Campbell's theorem then yields

$$P[N^*(t) > n] = \lambda E \int_0^t I[N^0(t-s) = n]\,\mathrm{d}s = \lambda \int_0^t P[N^0(t-s) = n]\,\mathrm{d}s\,, = \lambda \int_0^t P[N^0(s) = n]\,\mathrm{d}s\,.$$

■

8.4 The Palm Distribution: Conditioning in a Point at the Origin

Given any time-stationary MPP Ψ, its *Palm* distribution (named after C. Palm) is defined by

$$\boldsymbol{Q}(\cdot) = \lambda^{-1} E\left[\sum_{j=0}^{N(1)} I(\Psi_{(j)} \in \cdot)\right] ,$$

and the mapping taking $P(\Psi \in \cdot)$ to $\boldsymbol{Q}(\cdot)$ is called the *Palm transformation*. From (8.15), it follows that, if Ψ is also ergodic, then $\boldsymbol{Q}$ is the same as the point-stationary distribution $\boldsymbol{P}^0$ [as defined in (8.5)]. If ergodicity does not hold, however, then $\boldsymbol{Q}$ and $\boldsymbol{P}^0$ are different (in general), but the Palm distribution still yields a point-stationary distribution and any version distributed as $\boldsymbol{Q}$ is called a *Palm version* of Ψ.

Similarly, if we start with any point-stationary MPP Ψ, we can define a time-stationary distribution by

$$\boldsymbol{H}(\cdot) = \lambda E\left[\int_0^{T_0} I(\Psi_s \in \cdot)\mathrm{d}s\right] ,$$

which under ergodicity agrees with $\boldsymbol{P}^*$, but otherwise does not (in general). This mapping is called the *Palm inverse transformation* because applying it to $\boldsymbol{Q}$ yields back the original time-stationary distribution $P(\Psi \in \cdot)$. Together the two formulas are called the *Palm inversion formulas*. It should be emphasized that only in the non-ergodic case does the distinction between $\boldsymbol{Q}$ and $\boldsymbol{P}^0$ (or $\boldsymbol{H}$ and $\boldsymbol{P}^*$) become an issue because only when ergodicity holds can $\boldsymbol{Q}$ be interpreted as a point average [as defined in (8.5)], so one might ask if there is some other intuitive way to interpret $\boldsymbol{Q}$. The answer is yes: if Ψ is time-stationary, then its Palm distribution $\boldsymbol{Q}$ can be interpreted as *the conditional distribution of Ψ given a point at the origin*:

Theorem 8.4

If Ψ is time-stationary, then the Palm distribution Q can be obtained as the limiting distribution

$$\boldsymbol{Q}(\cdot) = \lim_{t\to 0} P(\Psi \in \cdot \mid t_0 \le t) ,$$

in the sense of weak convergence. Total variation convergence is obtained if Ψ is first shifted to t_0:

$$\boldsymbol{Q}(\cdot) = \lim_{t\to 0} P(\Psi_{(0)} \in \cdot \mid t_0 \le t) ,$$

in total variation.

As an immediate consequence, we conclude that (under ergodicity)

$$P(\Psi^0 \in \cdot) = \lim_{t\to 0} P(\Psi^* \in \cdot \mid t_0^* \le t)$$

(weak convergence) ,

$$P(\Psi^0 \in \cdot) = \lim_{t\to 0} P(\Psi^*_{(0)} \in \cdot \mid t_0 \le t)$$

(total variation convergence) .

Under ergodicity $\boldsymbol{P}^0$ can be viewed as the conditional distribution of $\boldsymbol{P}^*$ given a point at the origin.

A proof of such results can be carried out using inversion formulas and Khintchine–Korolyuk's Theorem 8.8.1 given in the next section which asserts that $P[N^*(t) > 0] \approx \lambda t$ as $t \to 0$.

Putting the one-sided renewal process aside, it is not true in general that $\Psi^*_{(0)}$ has a point-stationary distribution: shifting a time-stationary MPP to its initial point does not in general make it point-stationary; conditioning on $\{t_0^* \le t\}$ and taking the limit as $t \to 0$ is needed. [Recall the cyclic deterministic example in (8.12), for example.]

8.5 The Theorems of Khintchine, Korolyuk, and Dobrushin

For a Poisson process with rate λ, $P[N(t) = n] = \frac{\mathrm{e}^{-\lambda t}(\lambda t)^n}{n!}$, $n \in \mathbb{Z}_+$; thus $P[N(t) > 0] = 1 - \mathrm{e}^{-\lambda t}$ yielding (by L'Hospital's rule for example)

$$\lim_{t\to 0} \frac{P[N(t) > 0]}{t} = \lambda . \tag{8.20}$$

Similarly, $P[N(t) > 1] = 1 - \mathrm{e}^{-\lambda t}(1 + \lambda t)$ yielding

$$\lim_{t\to 0} \frac{P[N(t) > 1]}{t} = 0 . \tag{8.21}$$

Both (8.20) and (8.21) remain valid for any simple time-stationary point process, and the results are attributed to A. Y. Khintchine, V. S. Korolyuk, and R. L. Dobrushin. Any point process satisfying (8.21) is said to be *orderly*.

Theorem 8.5 (Khintchine–Korolyuk)
If Ψ is time stationary (and simple), then (8.20) holds.

Theorem 8.6 (Dobrushin)
If Ψ is time stationary (and simple), then (8.21) holds.

Proofs can easily be established using inversion formulas. For example, assume ergodicity and let $\Psi^* = \Psi$ with Ψ^0 being a point-stationary version with $F(x) = P(T^0 \le x)$ and $F_e(x) = \lambda \int_0^x [1 - F(y)]\,dy$. Then $P[N^*(t) > 0] = P(t_0^* \le t) = F_e(t)$, from the inversion formula (8.14). L'Hospital's rule then reduces the limit in (8.20) to $\lim_{t\to 0} \lambda[1 - F(t)] = \lambda$ [$F(0) = 0$ by simplicity]. Equation (8.21) can be proved from the Palm–Khintchine formula (8.19) for $n = 1$:

$$\begin{aligned} P[N^*(t) > 1] &= \lambda \int_0^t P[N^0(s) = 1]\,ds \\ &= \lambda \int_0^t P\left(t_1^0 \le s, t_2^0 > s\right) ds \\ &= \lambda \int_0^t P\left(t_1^0 \le s, t_2^0 > s\right) ds \\ &\le \lambda \int_0^t P\left(t_1^0 \le s\right) ds \le \lambda t F(t)\,; \end{aligned}$$

the result then follows since $F(0) = 0$ by simplicity.

8.6 An MPP Jointly with a Stochastic Process

In many applications an MPP Ψ is part of or interacts with some stochastic process $\boldsymbol{X} = [X(t) : t \ge 0]$, forming a joint process $(\boldsymbol{X}, \Psi)$. For example, Ψ might be the arrival times and service times to a queueing model, and $X(t)$ the state of the queue at time t. To accommodate this it is standard to assume that the sample paths of $\boldsymbol{X}$ are functions $x : \boldsymbol{R}_+ \to \boldsymbol{S}$ in the space

$$\mathcal{D}_S[0,\infty) \stackrel{\text{def}}{=} \{x :\ x \text{ is continuous from the right and has left-hand limits}\}\,,$$

endowed with the *Skorohod* topology. The state-space $\boldsymbol{S}$ can be a general complete separable metric space, but in many applications $\boldsymbol{S} = \boldsymbol{R}$, or a higher-dimensional Euclidean space. $\mathcal{D}_S[0,\infty)$ is denoted by $\mathcal{D}$ for simplicity.

Continuous from the right means thats for each $t \ge 0$: $x(t+) \stackrel{\text{def}}{=} \lim_{h\downarrow 0} x(t+h) = x(t)$, while *has left-hand limits* means that for each $t > 0$: $x(t-) \stackrel{\text{def}}{=} \lim_{h\downarrow 0} x(t-h)$ exits (and is finite). Such functions are also called *cadlag* (continue à droit, limits à gauchee) from the French.

It can be shown that such a function has, at most, countably many discontinuities, and is bounded on any finite interval $[a, b]$: $\sup_{t\in[a,b]} |x(t)| < \infty$. If t is a discontinuity, then the *jump* of X at t is defined by $x(t+) - x(t-)$.

Jointly the sample paths are pairs $(x, \psi) \in \mathcal{D} \times \boldsymbol{M}$ and this canonical space is endowed with the product topology and corresponding Borel sets. $(\boldsymbol{X}, \Psi) : \Omega \to \mathcal{D} \times \boldsymbol{M}$ formally is a mapping into the canonical space under some probability P; its distribution is denoted by $\boldsymbol{P}(\cdot) = P[(\boldsymbol{X}, \Psi) \in \cdot]$. The shifts θ_s and $\theta_{(j)}$ extend to this framework in a natural way by defining $\boldsymbol{X}_s = \theta_s \boldsymbol{X} = [X(s+t) : t \ge 0]$; $\theta_s(\boldsymbol{X}, \Psi) = (\boldsymbol{X}_s, \Psi_s)$. The notions of point and time stationarity (and ergodicity) go right through as does the notion of asymptotic stationarity, and the inversion formulas also go through. For example, the functional form of the inversion formulas in the canonical framework are:

$$\boldsymbol{E}^0[f(\boldsymbol{X}, \Psi)] = \lambda^{-1} \boldsymbol{E}^* \left[\sum_{j=0}^{N(t)} f(\boldsymbol{X}_{(j)}, \Psi_{(j)}) \right], \tag{8.22}$$

$$\boldsymbol{E}^*[f(\boldsymbol{X}, \Psi)] = \lambda \boldsymbol{E}^0 \left[\int_0^{T_0} f(\boldsymbol{X}_s, \Psi_s)\,ds \right]. \tag{8.23}$$

A point-stationary version is denoted by $(\boldsymbol{X}^0, \Psi^0)$, and has the property that $\boldsymbol{X}^0$ can be broken up into a stationary sequence of cycles $\mathcal{C}_n = [X^0(t_n^0 + t) : 0 \le t < T_n^0]$, $n \in \boldsymbol{Z}_+$, with cycle lengths being the interevent times $\{T_n\}$.

A time-stationary version is denoted by $(\boldsymbol{X}^*, \Psi^*)$, and $\boldsymbol{X}^*$ is a stationary stochastic process.

The two-sided framework goes through by letting $x : \boldsymbol{R} \to \boldsymbol{S}$ and using the extended space $\mathcal{D}(-\infty, +\infty)$.

8.6.1 Rate Conservation Law

Given a asymptotically stationary (and ergodic) pair $(\boldsymbol{X}, \Psi)$, with $\boldsymbol{X}$ real-valued, assume also that the sample paths of $\boldsymbol{X}$ are right differentiable, $x'(t) = \lim_{h \downarrow 0}[X(t+h) - x(t)]/x(t)$ exists for each t. Further assume that the points t_n of Ψ include all the discontinuity points (jumps) of $\boldsymbol{X}$ (if any); if for some t it holds that $X(t-) \neq X(t+)$, then $t = t_n$ for some n. Noting that (wp1) $\boldsymbol{E}^*[X'(0)] = \lim_{t\to\infty} \frac{1}{t}\int_0^t X'(s)\,\mathrm{d}s$ and $\boldsymbol{E}^0[X(0+) - X(0-)] = \lim_{n\to\infty} \frac{1}{n}\sum_{j=1}^n [X(t_j+) - X(t_j-)]$, average jump size, the following is known as *Miyazawa's rate conservation law* (RCL):

Theorem 8.7

If $\boldsymbol{E}^*|X'(0)| < \infty$ and $\boldsymbol{E}^0|X(0-) - X(0+)| < \infty$, then

$$\boldsymbol{E}^*(X'(0)) = \lambda \boldsymbol{E}^0[X(0-) - X(0+)]\,.$$

The time-average right derivative equals the arrival rate of jumps multiplied by the (negative of) the average jump size.

As an easy example, for $x \geq 0$ let $X(t) = [A(t) - x]_+$, where $A(t)$ is the forward recurrence time for Ψ. Then $A'(t) = -1$ and $X'(t) = -I[A(t) > x]$. Jumps are of the form $X(t_n+) - X(t_n-) = (T_n - x)_+$. The RCL then yields $P[A^*(0) > x] = \lambda E(T_0^0 - x)_+ = 1 - F_e(x)$. The RCL has many applications in queueing theory. For example consider Example 6 from Sect. 8.1.8 and let $X(t) = V^2(t)$. Then $V'(t) = -I[V(t) > 0]$ so $X'(t) = -2V(t)$ and $X(t_n+) - X(t_n-) = 2S_n D_n + S_n^2$; the RCL thus yields *Brumelle's formula*, $E(V) = \lambda E(SD) + \lambda E(S^2)/2$. (Here $SD = S_0^0 D_0^0$.) A sample-path version of the RCL can be found in [8.1].

8.7 The Conditional Intensity Approach

Motivated by the fact that $\{N(t) - \lambda t : t \geq 0\}$ forms a mean-zero martingale for a time-homogenous Poisson process with rate λ, the *conditional intensity* $\lambda(t)$ of a point process (when it exists) satisfies the property that $\{N(t) - \int_0^t \lambda(s)\,\mathrm{d}s\}$ forms a mean-zero martingale. The framework requires a history $\mathcal{F}_t$ supporting $N(t)$ and a heuristic definition is then $\lambda(t)\,\mathrm{d}t = E(N(\mathrm{d}t) \mid \mathcal{F}_t)$ which asserts that for each t the conditional expected number of new arrivals in the next $\mathrm{d}t$ time units, conditional on the history up to time t, is equal to $\lambda(t)\,\mathrm{d}t$. For a time-homogenous Poisson process at rate λ, $\lambda(t) = \lambda$; $E[N(\mathrm{d}t) \mid \mathcal{F}_t] = \lambda\,\mathrm{d}t$ due to stationary and independent increments; but for general point processes, $\lambda(t)$ (if it exists) depends on the past evolution (before time t).

A *non-stationary* Poisson process is a simple and very useful example, where the arrival rate λ changes over time, but $N(t)$ still has a Poisson distribution. A common example of this is when $\lambda(t)$ is a deterministic alternating function [e.g., $\lambda(t) = 2$ during the first 12 hours of each day, and $\lambda(t) = 1$ during the second 12 hours]. Intuitively then, a point process with an intensity is a generalization of a non-stationary Poisson process allowing for more complicated correlations over time.

Given any MPP Ψ, if $E[N(t)] < \infty$, $t \geq 0$, then $\{N(t)\}$ is always a non-negative right-continuous submartingale (with respect to its internal history), so the *Doob–Meyer decomposition* yields a right-continuous (and predictable) increasing process $\Lambda(t)$ (called the *compensator*) for which $\{N(t) - \Lambda(t)\}$ forms a mean-zero martingale. If $\Lambda(t)$ is of the form $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$, $t \geq 0$, where $\lambda(t)$ satisfies the regularity conditions of being non-negative, measurable, adapted to $\mathcal{F}_t$ and locally integrable [$\int_A \lambda(s)\,\mathrm{d}s < \infty$ for all bounded sets A], then $\lambda(t)$ is called the conditional intensity of the point process, or the *intensity* for short. (A predictable version of the intensity can always be chosen; this is done so by convention.) By the martingale property, an intensity can equivalently be defined as a stochastic process $\{\lambda(t)\}$ that satisfies the aforementioned regularity conditions and satisfies for all $s \leq t$

$$E[N(s,t] \mid \mathcal{F}_s] = E\left[\int_s^t \lambda(u)\,\mathrm{d}u \mid \mathcal{F}_s\right]\,.$$

Not all point processes admit an intensity. For example, a deterministic renewal process does not admit an intensity. The only part of $\mathcal{F}_t$ that is relevant for predicting the future of a renewal process is the backwards recurrence time $B(t)$, and if the interarrival time distribution F has a density f, then the renewal process admits an intensity $\lambda(t) = f[B(t-)]/\bar{F}[B(t-)]$, the hazard rate function of F evaluated at $B(t-)$. The fact that a density is needed illustrates the general fact that the existence of

an intensity requires some smoothness in the distribution of points over time.

Incorporating marks into an intensity amounts to making rigorous the heuristic $\lambda(t, \mathrm{d}k)\mathrm{d}t = E[\Psi(\mathrm{d}t \times \mathrm{d}k)|\mathcal{F}_t]$, for some *intensity kernal* $\lambda(t, \mathrm{d}k)$ which in integral form becomes

$$E\Big[\int H(t,k)\Psi(\mathrm{d}t \times \mathrm{d}k)\Big] = E\Big[\int H(t,k)\lambda(t, \mathrm{d}k)\Big],$$

for non-negative and predictable H.

Here $\lambda(t, \mathrm{d}k)$ is a measure on the mark space for each t. Equivalently such an intensity kernel must have the properties that, for each mark set K, the process $\{\lambda(t, K) : t \geq 0\}$ is adapted to $\{\mathcal{F}_t\}$ and serves as an intensity for the thinned point process (defined by its counting process) $N_K(t) = \Psi[(0, t] \times K]$.

An elementary example is given by the compound Poisson process at rate λ with (independent of its points) iid jumps $k_n = X_n$ with some distribution $\mu(\mathrm{d}x) = P(X \in \mathrm{d}x)$. Then $\lambda(t, \mathrm{d}x) = \lambda\mu(\mathrm{d}x)$.

8.7.1 Time Changing to a Poisson Process

In some applications, it is desirable to construct (or simulate) a point process with a given intensity or corresponding compensator $\Lambda(t) = \int_0^t \lambda(s)\mathrm{d}s$. This can generally be accomplished by defining $N(t) = M[\Lambda(t)]$, where $M(t)$ is the counting process for an appropriate time-homogenous Poisson process at rate $\lambda = 1$. Conversely, the Poisson process can be retrieved by inverting the time change; $M(t) = N[\Lambda^{-1}(t)]$.

Theorem 8.8

Consider the counting process $\{N(t)\}$ of a (simple) MPP with intensity $\{\lambda(t)\}$ that is strictly positive and bounded. [Also assume that $\Lambda(t) \to \infty$, as $t \to \infty$, wp1.] Then $M(t) \stackrel{\text{def}}{=} N[\Lambda^{-1}(t)]$ defines a time-homogenous Poisson process at rate $\lambda = 1$.

There are some extensions of this result that incorporates the marks, in which case the time-homogenous Poisson process is replaced by a compound Poisson process.

8.7.2 Papangelou's Formula

Suppose that Ψ is asymptotically stationary (and ergodic), and that Ψ^* (two-sided) admits a conditional intensity $\lambda(t)$ with respect to a history $\mathcal{F}_t$.

Proposition 8.4 (Papangelou's formula)

For all non-negative random variables $X \in \mathcal{F}_{0-} \stackrel{\text{def}}{=} \cup_{t<0} \mathcal{F}_t$

$$\boldsymbol{E}^*[\lambda(0)X] = \lambda \boldsymbol{E}^0(X)\,.$$

In other words, conditional on $\mathcal{F}_{0-}$, $\boldsymbol{P}^0$ is absolutely continuous with respect to $\boldsymbol{P}^*$, $\boldsymbol{P}^0 \ll \boldsymbol{P}^*$, having Radon–Nikodým derivative

$$\frac{\mathrm{d}\boldsymbol{P}^0}{\mathrm{d}\boldsymbol{P}^*}\,|\mathcal{F}_{0-} = \frac{\lambda(0)}{\lambda}\,.$$

Note that, when $X = 1$, the basic fact that $\boldsymbol{E}^*(\lambda(t)) = \lambda$ is retrieved. In many applications, $\mathcal{F}_t$ supports a stochastic process $\boldsymbol{X}$ jointly with Ψ in which case, letting $X = X(0-)$, Papangelou's formula yields $\boldsymbol{E}^*[\lambda(0)X(0-)] = \lambda \boldsymbol{E}^0[X(0-)]$. For example in the single-server queue Example 6 from Sect. 8.1.8, letting $X(t) = I[V(t) \leq x]$ yields $\boldsymbol{E}^*\{\lambda(0)I[V(0-) \leq x]\} = \lambda \boldsymbol{P}^0[V(0-) \leq x] = \lambda P(D \leq x)$. If arrivals are Poisson $[\lambda(t) = \lambda]$, this reduces to $P(V \leq x) = P(D \leq x)$ illustrating *Poisson arrivals see time averages* (PASTA), which asserts that when arrival times are Poisson for a queueing process $\{X(t)\}$ and satisfy a lack of anticipation condition (LAC), then the distribution of $X(0-)$ is the same under $\boldsymbol{P}^0$ and $\boldsymbol{P}^*$. LAC is defined as: for each $t \geq 0$ the future increments $\{N(t+s) - N(t) : s \geq 0\}$ are independent of the joint past $\{\{X(s) : s \leq t\}, \{N(s) : s \leq t\}\}$. Recalling the definition of $\boldsymbol{P}^0$ and $\boldsymbol{P}^*$ as Cesàro averages, PASTA says that if arrivals are Poisson then

the proportion of arrivals who find the queueing process in a given state is equal to the proportion of time the system is in that state;

$$\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} I[X(t_j-) \in A] = \lim_{t\to\infty} \frac{1}{t} \int_0^t I[X(s) \in A]\mathrm{d}s\,.$$

Although its origins and primary applications are in queueing theory, PASTA can be applied to any joint pair $(\boldsymbol{X}, \Psi)$ for which the points of Ψ form a Poisson process satisfying LAC.

8.8 The Non-Ergodic Case

The assumption of ergodicity can be relaxed to allow for a stationary but non-ergodic framework. The key is in conditioning first on the *invariant σ-field* of $\mathbb{M}$,

$$\Im \stackrel{\text{def}}{=} \{\mathcal{E} \subset \mathbb{M} : \theta_t^{-1}\mathcal{E} = \mathcal{E},\ t \geq 0\}\,,$$

where $\theta_t^{-1}\mathcal{E} = \{\psi \in \mathbb{M} : \psi_t \in \mathcal{E}\}$ and the $\mathcal{E}$ are restricted to be Borel sets. Ergodicity is the case when $\boldsymbol{P}(\mathcal{E}) = 0$ or 1 for all $\mathcal{E} \in \Im$. Conditional expectation is denoted by $\boldsymbol{E}_{\Im}(X) \stackrel{\text{def}}{=} \boldsymbol{E}(X|\ \Im)$, and so on. A typical example of an invariant set is $\mathcal{E}_2 = \{\psi \in \mathbb{M} :\ \lim_{t\to\infty} N(t)/t = 2\}$, the set of all ψ with arrival rate exactly 2; it follows immediately that $\theta_t^{-1}\mathcal{E}_2 = \mathcal{E}_2$ because shifting a point process does not effect its long-term arrival rate.

A typical example of a stationary but non-ergodic MPP is a mixture of two Poisson processes, one at rate 1 and one at rate 2: flip a fair coin once. If it lands heads, then choose Ψ as a Poisson process at rate 1; if it lands tails, then choose Ψ as a Poisson process at rate 2. In this case $\boldsymbol{P}(\mathcal{E}_2) = 0.5$, Ψ is non-ergodic.

Inversion formulas still hold between Ψ^* and Ψ^0 if one conditions on $\Im$ first. For example, letting $\lambda_{\Im} \stackrel{\text{def}}{=} \boldsymbol{E}^*_{\Im}[N(1)]$, it holds that

$$P(t_0^* > x) = \boldsymbol{E}^0\{\lambda_{\mathrm{i}} \boldsymbol{E}^0_{\Im}(T_0 - x)_+\}\,. \tag{8.24}$$

In the above Poisson process case, $\lambda_{\Im} = 1$ if the coin lands heads, or 2 if it lands tails, and (8.24) reduces to

$$P(t_0^* > x) = \frac{(\mathrm{e}^{-x} + \mathrm{e}^{-2x})}{2}\,.$$

If the mixture was for two renewal processes with interarrival time distributions F_1 and F_2 respectively, then (8.24) reduces to

$$P(t_0^* > x) = \frac{[\overline{F}_{1,e}(x) + \overline{F}_{2,e}(x)]}{2}\,,$$

involving the two stationary excess distributions. The general inversion formula from $\boldsymbol{P}^0$ to $\boldsymbol{P}^*$ in functional form becomes

$$\boldsymbol{E}^*[f(\Psi)] = \boldsymbol{E}^0\left\{\lambda_{\mathrm{i}} \boldsymbol{E}^0_{\Im}\left[\int_0^{T_0} f(\Psi_s)\,\mathrm{d}s\right]\right\}\,.$$

8.9 MPPs in $\mathbb{R}^d$

When a point process has points in a higher-dimensional space such as $\mathbb{R}^d$, then the theory becomes more complicated. The main reason for this is that there is no longer a natural ordering for the points, e.g., there is no "next" point as is the case on $\mathbb{R}$. So "shifting to the j-th point" to obtain $\Psi_{(j)}$ is no longer well-defined. To make matters worse, point Cesàro limits as in (8.5) depend upon the ordering of the points. Whereas when $d = 1$ there is a one-to-one correspondence between stationary sequences of non-negative RVs (interarrival times) and point-stationary point processes, in higher dimensions such a simple correspondence is elusive. A good example to keep in mind is mobile phone usage, where the points (in $\mathbb{R}^2$ for simplicity) denote the locations of mobile phone users at some given time, and for each user the marks might represent whether a phone call is in progress or not. As in one dimension, it would be useful to consider analyzing this MPP from two perspectives: from the perspective of a "typical" user, and from the perspective of a "typical" spatial position in $\mathbb{R}^2$. For example, one might wish to estimate the average distance from a typical user to a base station, or the average distance from a typical position to a user with a call in progress. A mobile phone company trying to decide where to place some new base stations would benefit by such an analysis.

Some of the multidimensional complications can be handled, and initially it is best to use the measure approach from Sect. 8.1.5 to define an MPP. Starting with $\psi = \{(x_j, k_j)\}$, where $x_j \in \mathbb{R}^d$, it can equivalently be viewed as a σ-finite $\mathbb{Z}_+$-valued measure

$$\psi = \sum_j \delta_{(x_j,k_j)}\,,$$

on (the Borel sets of) $\mathbb{R}^d \times \mathbb{K}$.

The counting process is replaced by the counting measure $N(A) =$ the number of points that fall in the Borel set $A \subset \mathbb{R}^d$, and it is assumed that $N(A) < \infty$ for all bounded A. Simple means that the points x_j are distinct; $N(\{x\}) \leq 1$ for all $x \in \mathbb{R}^d$.

For any x, the shift mapping $\theta_x\psi = \psi_x$ is well defined via $\psi_x(A \times K) = \psi(A + x,\, K)$, where $A + x = \{y + x : y \in A\}$.

8.9.1 Spatial Stationarity in $\mathbb{R}^d$

Analogous to time stationarity in $\mathbb{R}$, the definition of *spatial stationarity* is that Ψ_x has the same distribution for all $x \in \mathbb{R}^d$, and as in (8.8) such MPPs can be viewed as arising as a Cesàro average over space, as follows.

Let B_r denote the d-dimensional ball of radius r centered at 0. Then (with l denoting Lebesgue measure in $\mathbb{R}^d$) a spatially stationary MPP is obtained via

$$P(\Psi^* \in \cdot) \stackrel{\text{def}}{=} \lim_{r\to\infty} \frac{1}{l(B_r)} \int_{B_r} P(\Psi_x \in \cdot)\,\mathrm{d}x \,.$$

In essence, we have randomly chosen our origin from over all of space.

Ergodicity means that the flow of shifts $\{\theta_x\}$ is ergodic. Stationarity implies that $E[N(A)] = \lambda l(A)$ for some λ, called the *mean density*; it can be computed by choosing (say) A as the unit hypercube $H = [0,1]^d$; $\lambda = E[N(H)]$, the expected number of points in any set of volume 1.

An important example in applications is the Poisson process in $\mathbb{R}^d$. $N(A)$ has a Poisson distribution with mean $\lambda l(A)$ for all bounded Borel sets A, and $N(A_1)$ and $N(A_2)$ are independent if $A_1 \cap A_2 = \emptyset$.

8.9.2 Point Stationarity in $\mathbb{R}^d$

Coming up with a definition of point stationarity, however, is not clear, for what do we mean by "randomly selecting a point as the origin", and even if we could do just that what stationarity property would the resulting MPP have? (For example, even for a spatially stationary two-dimensional Poisson process, if a point is placed at the origin, it is not clear in what sense such a point process is stationary.) One would like to be able to preserve the distribution under a point shift, but which point can be chosen as the one to shift to as the new origin? Under ergodicity, one could define $P(\Psi^0 \in \cdot)$ as a sample-path average

$$P(\Psi^0 \in \cdot) \stackrel{\text{def}}{=} \lim_{r\to\infty} \frac{1}{N(B_r)} \sum_{x\in B_r} I(\Psi_x \in \cdot), \ \text{wp1} \,.$$

It turns out that this can be improved to be more like (8.5) as follows. Let p_n denote the n-th point hit by B_r as $r \to \infty$ (if there are ties just order lexicographically). For each sample path of Ψ, $\{p_n\}$ is a permutation of $\{x_n\}$. Define

$$P(\Psi^0 \in \cdot) \stackrel{\text{def}}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} P(\Psi_{p_j} \in \cdot) \,.$$

Another approach involves starting with the spatially stationary MPP Ψ^* and defining $P(\Psi^0 \in \cdot)$ by inversion in the spirit of (8.15) and the Palm transformation, replacing a "unit of time" by any set A with volume 1, such as the unit hypercube $H = (0,1]^d$:

$$P(\Psi^0 \in \cdot) = \lambda^{-1} E\left[\sum_{x\in H} I(\Psi_x^* \in \cdot)\right] \,. \tag{8.25}$$

Under ergodicity all these methods yield the same distribution. Ψ^0 has the property that there is a point at the origin, and its distribution is invariant under a two-step procedure involving an external randomization followed by a random point shift as follows (see Chapt. 9 of *Thorisson* [8.2]):

> *First, randomly place a ball B_r of any fixed radius $r > 0$ over the origin, e.g., take U distributed uniformly over the open ball B_r and consider the region $R = B_r + U$. There is at least one point in R, the point at the origin, but in any case let $n = N(R)$ denote the total number. Second, randomly choose one of the n points (e.g., according to the discrete uniform distribution) and shift to that point as the new origin. This shifted* MPP *has the same distribution $P(\Psi^0 \in \cdot)$ as it started with.*

A recent active area of research is to determine whether or not one can achieve this invariance without any randomization. In other words is there an algorithm for choosing the "next point" to move to only using the sample paths of Ψ^0? In one dimension we know this is possible; always choose (for example) the point to the right (or left) of the current point. It turns out that in general this can be done (*Heveling* and *Last* [8.3]), but what is still not known is whether it can be done in such a way that all the points of the point process are exhaustively visited if the algorithm is repeated (as is the case in one dimension). For the Poisson process with $d = 2$ or 3 simple algorithms have indeed been found (*Ferrari* et al. [8.4]).

8.9.3 Inversion and Voronoi Sets

There is an analogue for the inverse part of the formula (8.25) in the spirit of (8.14), but now there is no "cycle" to average over so it is not clear what to do. It turns out that a random *Voronoi cell* is needed. For an MPP ψ with points $\{x_j\}$, for each point x_i define the

Voronoi cell about x_i by

$$V_{x_i}(\psi) = \{x \in \boldsymbol{R}^d : ||x - x_i|| < ||x - x_j|| \text{ , for all points } x_j \neq x_i\} ,$$

the set of elements in $\boldsymbol{R}^d$ that are closer to the point x_i than they are to any other point of ψ. For an MPP, this set is a random set containing x_i and of particular interest is when $\Psi = \Psi^0$ and $x_i = 0$, the point at the origin. We denote this Voronoi cell by V_0. It turns out that $E[l(V_0)] = \lambda^{-1}$, and

$$P(\Psi^* \in \cdot) = \lambda E\left[\int_{V_0} I\left(\Psi_x^0 \in \cdot\right) \mathrm{d}x\right] . \tag{8.26}$$

The Voronoi cell V_0 plays the role that the interarrival time $T_0^0 = t_1^0$ does when $d = 1$. But, even when $d = 1$, V_0 is not the same as an interarrival time; instead it is given by the random interval $V_0 = (-T_{-1}^0/2, T_0^0/2) = (t_{-1}^0/2, t_1^0/2)$ which has length $l(V_0) = (t_1^0 + |t_{-1}^0|)/2$ and hence mean λ^{-1}. It is instructive to look closer at this for a Poisson process at rate λ, for then $l(V_0)$ has an Erlang distribution with mean λ^{-1}.

In the mobile phone context, if the points x_i are now the location of base stations (instead of phones) then V_{x_i} denotes the service zone for the base station, the region about x_i for which x_i is the closest base station. Any mobile user in that region would be best served (e.g., minimal distance) by being connected to the base at x_i. Thus all of space can be broken up into a collection of disjoint service zones corresponding to the Voronoi cells.

Finally, analogous to the $d = 1$ case, starting with a spatially stationary MPP it remains valid (in a limiting sense as in Theorem 8.8.4) that the distribution of Ψ^0 can be obtained as the conditional distribution of Ψ^* given a point at the origin. For example, placing a point at the origin for a spatially stationary Poisson process Ψ^* in $\mathbb{R}^d$ yields Ψ^0.

References

8.1 K. Sigman: *Stationary Marked Point Processes: An Intuitive Approach* (Chapman Hall, New York 1995)

8.2 H. Thorisson: *Coupling, Stationarity, and Regeneration* (Springer, Heidelberg Berlin New York 2000)

8.3 M. Heveling, G. Last: Characterization of Palm measures via bijective point-shifts, Annals of Probability **33**(5), 1698–1715 (2004)

8.4 P. A. Ferrari, C. Landim, H. Thorisson: Poisson trees, succession lines and coalescing random walks, Annals de L'Institut Henry Poincaré **40**, 141–152 (2004)

9. Modeling and Analyzing Yield, Burn-In and Reliability for Semiconductor Manufacturing: Overview

The demand for proactive techniques to model yield and reliability and to deal with various infant mortality issues are growing with increased integrated circuit (IC) complexity and new technologies toward the nanoscale. This chapter provides an overview of modeling and analysis of yield and reliability with an additional burn-in step as a fundamental means for yield and reliability enhancement.

After the introduction, the second section reviews yield modeling. The notions of various yield components are introduced. The existing models, such as the Poisson model, compound Poisson models and other approaches for yield modeling, are introduced. In addition to the critical area and defect size distributions on the wafers, key factors for accurate yield modeling are also examined. This section addresses the issues in improving semiconductor yield including how clustering may affect yield.

The third section reviews reliability aspects of semiconductors such as the properties of failure mechanisms and the typical bathtub failure rate curve with an emphasis on the high rate of early failures. The issues for reliability improvement are addressed.

The fourth section discusses several issues related to burn-in. The necessity for and effects of burn-in are examined. Strategies for the level and type of burn-in are examined. The literature on optimal burn-in policy is reviewed. Often percentile residual life can be a good measure of performance in addition to the failure rate or reliability commonly used.

The fifth section introduces proactive methods of estimating semiconductor reliability from yield information using yield–reliability relation models. Time-dependent and -independent models are discussed.

The last section concludes this chapter and addresses topics for future research and development.

Since Jack Kilby of Texas Instruments invented the first integrated circuit (IC) in 1958, the semiconductor industry has consistently developed more complex chips at ever decreasing cost. Feature size has shrunk by 30% and die area has grown by 12% every three years [9.1]. The number of transistors per chip has grown exponentially while semiconductor cost per function has been reduced at the historical rate of 25% per year. As shown in Table 9.1, the semiconductor market will reach $213 billion in 2004, which represents 28.5% growth over

Table 9.1 Industry sales expectations for IC devices [9.2]

Device type	Billion dollars				Percent growth			
	2003	2004	2005	2006	03/02	04/03	05/04	06/05
Discretes	13.3	16.0	17.0	16.7	8.1	20.2	6.2	−2.0
Optoelectronics	9.5	13.1	14.9	15.3	40.6	37.3	13.4	2.9
Actuators	3.5	4.8	5.7	6.3	[a]	35.3	18.9	9.1
Bipolar digital	0.2	0.2	0.2	0.2	−4.2	10.6	−16.3	−25.0
Analog	26.8	33.7	37.0	37.0	12.0	25.6	9.9	−0.1
MOS micro	43.5	52.4	57.2	57.6	14.3	20.4	9.2	0.6
MOS logic	36.9	46.4	50.6	49.6	18.1	25.7	9.1	−2.1
MOS memory	32.5	46.9	49.1	47.6	20.2	44.4	4.6	−3.1
Total	166.4	213.6	231.7	230.0	18.3	28.5	8.5	−0.7

[a] A growth rate is not meaningful to show since WSTS included actuators from 2003

2003. Growth of 8.5% is forecasted for 2005, followed by virtually zero growth in 2006. In 2007, however, another recovery cycle is expected to begin with market growth in the 10% range.

Clearly, yield and reliability are two of the cornerstones of successful IC manufacturing as they measure semiconductor facility profitability and post-manufacturing device failures. Yield and reliability have played a key role in many aspects of semiconductor operations such as determining the cost of new chips under development, forecasting time-to-market, defining the maximum level of integration possible and estimating the number of wafers to start with. Traditionally, reactive techniques have been used to analyze yield and reliability, and an investigation was launched to determine the cause of yield loss once a low yield was observed during production. Stress testing and failure analysis were commonly performed at the end of the manufacturing line [9.3, 4]. However, as the rapid increase in IC complexity has resulted in multi-billion-dollar semiconductor fabrication facilities, IC manufacturers struggle to obtain a better return on their investment by introducing new process technologies and materials at an accelerated rate to satisfy narrowing market windows. Given this trend, the demand for proactive techniques has strengthened in order to achieve the desired yield and reliability goals early in the process or even before production begins. The demand for these proactive techniques will be even bigger in emerging nanotechnology, which is known to have low yield and reliability [9.5, 6].

Yield and reliability modeling and analysis is a means of achieving proactive yield and reliability management. The purpose of this paper is to review the modeling and analysis of yield and reliability with an additional burn-in step. The importance of yield modeling is emphasized for obtaining better yields quickly after new technologies are introduced. In particular, the relationship between yield, burn-in and reliability will be thoroughly addressed. The relation model between yield and reliability can aid in design for manufacturability (DFM) by improving device layouts for better manufacturing yield and reliability during their early development prior to manufacturing.

9.1 Semiconductor Yield

Yield in semiconductor technology is the most important index for measuring success in the IC business. In general, yield is defined as the fraction of manufactured devices that meet all performance and functionality specifications. Higher yield tends to produce more chips at the same cost, thus allowing prices to decrease.

In this section, we first decompose overall yield into several components. Then, the literature on yield models is reviewed, focusing mainly on the random defect yield model. Traditional Poisson and compound Poisson yield models are thoroughly reviewed as well as some more recent yield models. Finally, issues related to proactive

yield improvement are discussed from the viewpoint of yield modeling.

9.1.1 Components of Semiconductor Yield

The overall yield Y_{overall} of a semiconductor facility can be broken down into several components: wafer process yield Y_{process}, wafer probe yield Y_{probe}, assembly yield Y_{assembly} and final test yield $Y_{\text{final test}}$ [9.7]. Wafer process yield, which is synonymous with line or wafer yield, is the fraction of wafers that complete wafer fabrication. Wafer probe yield is the fraction of chips on yielding wafers that pass the wafer probe test. The terms die yield, chip yield or wafer sort yield are used interchangeably with wafer probe yield. Overall yield is the product of these components, written as

$$Y_{\text{overall}} = Y_{\text{process}} Y_{\text{probe}} Y_{\text{assembly}} Y_{\text{final test}} .$$

9.1.2 Components of Wafer Probe Yield

Most semiconductor industries focus on improving the wafer probe yield, which is the bottleneck of overall yield. The importance of wafer probe yield to financial success is discussed in [9.8, 9]. Wafer probe yield is decomposed into functional yield $Y_{\text{functional}}$ and parametric yield $Y_{\text{parametric}}$ such that

$$Y_{\text{probe}} = Y_{\text{functional}} Y_{\text{parametric}} .$$

Parametric yield refers to the quantification of IC performance that is caused by process parameter variations. The designer attempts to increase parametric yield using several tools to check the design for process and parameter variations. Commonly used methods include corner analysis, Monte Carlo analysis, and the response surface methodology [9.10]. Corner analysis is the most widely used method due to its simplicity. The designer determines the worst-case corner under which the design can be expected to function. Then, each corner is simulated and the output is examined to ascertain whether or not the design performs as required. The disadvantages of corner analysis include the possibility that a design may function well at the corners but fail in between or that the designer may not know what the corners are. In Monte Carlo analysis, samples are generated to estimate yield based on the distributions of the process parameters. A disadvantage of Monte Carlo analysis is that the designer may not know if an increased yield is due to a change in the design parameters or is due to Monte Carlo sampling error. Another disadvantage is that a complete rerun of the analysis is required if the design variables are changed. With the response surface methodology, a set of polynomial models are created from the design of experiments that approximate the original design. These models are run so many times that the sampling error is reduced to nearly zero. A disadvantage of the response surface methodology arises from errors existing as a result of differences between the polynomial models and the original design.

Functional yield is related to manufacturing problems such as particulate matter, mechanical damage, and crystalline defects which cause dice not to function. Therefore, functional yield is a reflection of the quality of the manufacturing process and is often called the manufacturing yield [9.7, 11] or the catastrophic yield [9.12, 13]. In general, functional yield can be further partitioned into three categories: repeating yield $Y_{\text{repeating}}$, systematic yield $Y_{\text{systematic}}$ and random-defect-limited yield Y_{random} [9.14]:

$$Y_{\text{functional}} = Y_{\text{repeating}} Y_{\text{systematic}} Y_{\text{random}} .$$

Repeating yield is limited to reticle defects that occur when there are multiple dies on a reticle. Once reticle defects are identified using a pattern-recognition algorithm, repeating yield is calculated by the ratio of the number of dies without repeating defects to the total number of dies per wafer [9.9]. Then, repeating yield is extracted from functional yield, and tile yield is defined by

$$Y_{\text{tile}} = \frac{Y_{\text{functional}}}{Y_{\text{repeating}}} = Y_{\text{systematic}} Y_{\text{random}} .$$

Systematic yield is limited to nonrandom defects affecting every die in some region of the wafer. To decompose Y_{tile} into $Y_{\text{systematic}}$ and Y_{random}, $Y_{\text{systematic}}$ is assumed to be constant regardless of die size since, in a mature process, $Y_{\text{systematic}}$ is known and controllable and is often equal to one. Then, a model is selected to relate Y_{random} to the die area and the density of the random defects, and curve fitting is used with

$$\ln Y_{\text{tile}} = \ln Y_{\text{systematic}} + \ln Y_{\text{random}}$$

to estimate $Y_{\text{systematic}}$ and the parameters of a model for Y_{random} [9.9].

9.1.3 Modeling Random Defect Yield

Since the early 1960s, researchers have devoted extensive work to developing yield models that relate the mean number of random defects in a device to the device yield.

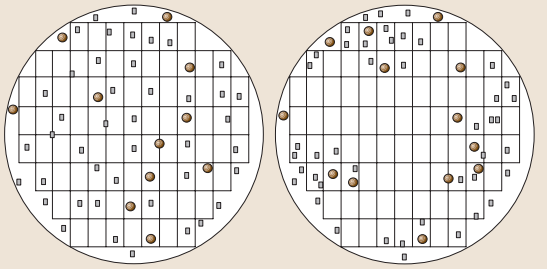


Fig. 9.1 Comparison of defect clustering for the same defect density [9.15]

The basic assumption is that yield is a function of the device area and the average number of yield defects per unit area.

During the manufacturing process, random defects can be introduced at any one of hundreds of process steps. Not all defects necessarily cause device failures. A defect that is of sufficient size and/or occurs in a place that results in an immediate device failure is called a fatal defect [9.16, 17], killer defect [9.18–22] or yield defect [9.23–27]. On the other hand, a defect that is either too small or located in a position that does not cause an immediate failure is called a latent defect, nonfatal defect or reliability defect. In this chapter, we will use the terms yield defect and reliability defect.

Poisson Model

For the purpose of yield modeling, the only yield defects that are of interest are those that can be detected by a manufacturing yield test. Let N_y be the number of yield defects introduced during fabrication on a device of area A. Assuming that the defects are randomly distributed and the occurrence of a defect at any location is independent of the occurrence of any other defect, the probability of a device having k yield defects is calculated by the Poisson probability distribution:

$$P_{\text{Poisson}}(k) = P(N_y = k) = \frac{\mathrm{e}^{-\lambda_y}\lambda_y^k}{k!}\,, \quad k = 0, 1, 2\,, \tag{9.1}$$

where λ_y is the average number of yield defects with $\lambda_y = E(N_y)$. Then, the corresponding Poisson yield is obtained by

$$Y_{\text{Poisson}} = P_{\text{Poisson}}(0) = \mathrm{e}^{-\lambda_y}\,, \tag{9.2}$$

where $\lambda_y = AD_y$, and D_y is the average number of yield defects per unit area.

Compound Poisson Model

Often defects on ICs are not uniformly distributed but tend to cluster. When defects are clustered in certain areas, the Poisson distribution is too pessimistic and the compound Poisson process is used, given by

$$P_{\text{compound}}(k) = P(N_y = k) = \int \frac{\mathrm{e}^{-AD}(AD)^k}{k!} f(D)\mathrm{d}D\,, \quad k = 0, 1, 2\,,$$

where $f(D)$ is the distribution of the defect density. The corresponding yield expression is

$$Y_{\text{compound}} = P_{\text{compound}}(0) = \int \mathrm{e}^{-AD} f(D)\mathrm{d}D\,.$$

Figure 9.1 compares two different degrees of defect clustering for the same average defect density. The left one, with low clustering, belongs more to the Poisson model and the right one, with high clustering, belongs more to the compound Poisson model.

Several distributions such as the symmetric triangle, exponential, uniform, gamma, Weibull and inverse Gaussian have been suggested for $f(D)$ [9.7, 16, 28, 29]. If D follows a uniform distribution in $[0, 2D_y]$, then the corresponding yield can be obtained by

$$Y_{\text{uniform}} = \int_0^{2D_y} \mathrm{e}^{-AD}\frac{1}{2D_y}\mathrm{d}D = \frac{1-\mathrm{e}^{-2\lambda_y}}{2\lambda_y}\,.$$

In the case where D follows a triangle distribution that approximates a normal distribution, the resulting model is called the Murphy's yield and is derived by

$$\begin{aligned} Y_{\text{Murphy's}} &= \int_0^{D_y} \mathrm{e}^{-AD}\frac{D}{D_y^2}\mathrm{d}D \\ &\quad + \int_{D_y}^{2D_y} \mathrm{e}^{-AD}\left(2-\frac{D}{D_y}\right)\frac{1}{D_y}\mathrm{d}D \\ &= \left(\frac{1-\mathrm{e}^{-\lambda_y}}{\lambda_y}\right)^2\,. \end{aligned}$$

For an exponential distribution of D, the model is called the Seed's yield and is given by

$$Y_{\text{Seed's}} = \int_0^{\infty} \mathrm{e}^{-AD}\frac{\mathrm{e}^{-D/D_y}}{D_y}\mathrm{d}D = \frac{1}{1+\lambda_y}\,.$$

For the case of the Weibull distribution, the corresponding yield is [9.29]

$$Y_{\text{Weibull}} = \int_0^\infty \mathrm{e}^{-AD}\frac{\alpha}{\beta^\alpha}D^{\alpha-1}\mathrm{e}^{-(D/\beta)^\alpha}\,\mathrm{d}D = \sum_{k=0}^{\infty}(-1)^k\frac{(AD_y)^k}{k!}\frac{\Gamma(1+k/\alpha)}{\Gamma^k(1+1/\alpha)}\,.$$

Also, if $f(D)$ is the inverse Gaussian distribution, then the yield is [9.29]

$$Y_{\text{inverse-Gaussian}} = \int_0^\infty \mathrm{e}^{-AD}\sqrt{\frac{\phi}{2\pi}}x^{-3/2}\exp\left(-\frac{\phi(D-D_y)^2}{2D_y^2D}\right)\mathrm{d}D = \exp\left\{\phi\left[1-\left(1+\frac{2AD_y}{\phi}\right)^{1/2}\right]\right\}\,.$$

When D follows a gamma distribution, the resulting model is called the negative binomial yield, which is derived as

$$Y_{\text{nb}} = \int_0^\infty \mathrm{e}^{-AD}\frac{1}{\Gamma(\alpha)\beta^\alpha}D^{\alpha-1}\mathrm{e}^{-D/\beta}\,\mathrm{d}D = \left(1+\frac{\lambda_y}{\alpha}\right)^{-\alpha}\,, \tag{9.3}$$

where α is referred to as the clustering factor. A smaller value of α means a higher degree of clustering and greater variation in defect density across the wafer. If $\alpha = 1$, then the negative binomial yield is equivalent to Seed's yield. In the case where $\alpha \to \infty$, the negative binomial yield approaches the Poisson yield. By varying the value of α, the negative binomial yield covers the whole range of yield estimations. *Cunningham* [9.28] reported methods to determine the clustering factor. *Langford* and *Liou* [9.30] presented a new technique to calculate an exact solution of α from wafer probe bin map data.

Critical Area and Defect Size Distribution in Yield Model

The random-defect-limited yield can be more accurately evaluated if the concepts of critical area and defect size distribution are incorporated.

Let $s(x)$ be the probability density function of the defect size. Although the form of $s(x)$ depends on process lines, process time, learning experience gained and other variables, it generally peaks at a critical size and then decreases on either side of the peak [9.31]. Let x_0 be the critical size of the defect that is most likely to occur. The defect size distribution is given by [9.7, 15, 23, 28]

$$s(x) = \begin{cases} cx_0^{-q-1}x^q, & 0 \le x \le x_0 \\ cx_0^{p-1}x^{-p}, & x_0 < x < \infty\,, \end{cases} \tag{9.4}$$

where $p \neq 1, q > 0$ and $c = (q+1)(p-1)/(p+q)$. While p, q and x_0 are process-dependent constants, $q = 1$ and $p = 3$ agree well with the experimental data, and x_0 must be smaller than the minimum width or spacing of the defect monitor [9.7, 23]. A gamma distribution is also used for $s(x)$ in some applications [9.32, 33].

The critical area defines the region of the layout where a defect must fall to cause device failure. Therefore, if a defect occurs in the critical area, then it becomes a yield defect. Given $s(x)$, the yield critical area is expressed by

$$A_y = \int_0^\infty A_y(x)s(x)\,\mathrm{d}x\,,$$

where $A_y(x)$ is a critical area of defect size x. Then $\lambda_y = A_yD_0$ is used in yield models where D_0 is the average defect density of all sizes. The geometric method [9.34], the Monte Carlo method [9.35] and the pattern-oriented method [9.36] have been used for critical area extraction. Critical area analysis can be used to quantify the sensitivity of a design to defects based on the layout [9.37] and can aid DFM by improving layouts for better yield [9.38].

Other Models

Sato et al. [9.39] used a discrete exponential distribution for the number of yield defects for each type of defect:

$$P_{\text{discrete expo}}(k) = (1-\mathrm{e}^{-h})\mathrm{e}^{-hk}\,, \tag{9.5}$$

where h is the parameter. Once the probability density function for m types of defects is derived by m convolution of (9.5), the yield is derived by

$$Y_{\text{discrete expo}} = (1-\mathrm{e}^{-h})^m = (1+A_yD_0/m)^{-m}\,. \tag{9.6}$$

Park and *Jun* [9.40] presented another yield model based on a generalized Poisson distribution. Assuming that the number of defect clusters in a chip and the number of defects in each cluster follows a Poisson distribution, the total number of defects in a chip follows a generalized Poisson distribution. Then yield is calculated using the

fact that the total number of yield defects in a chip is the sum of the yield defects in each cluster if the probability of each defect in the cluster becoming a yield defect is the same.

Jun et al. [9.41] developed a yield model through regression analysis in which the mean number of defects per chip and a new cluster index obtained from the defect location are used as independent variables. Simulation results showed that yield is higher for the higher index, but the rate of yield growth decreases as the cluster index increases.

Carrasco and *Suñé* [9.42] presented a methodology for estimating yield for fault-tolerant systems-on-chip, assuming that the system fails with probability $1 - C_i$ if component i fails. The system failure probability is independent of the subsets of components which failed before. For each component, an upper bound on the yield loss is obtained, which can be formalized as the probability that a Boolean function of certain independent integer-valued random variables is equal to one. The reduced-order multiple-valued decision diagram is used to compute the probability.

Noting that interconnect substrates face low yield and high cost, *Scheffler* et al. [9.43] presented a yield estimation approach to assess the impact on overall substrate cost of changing design rules. Given a defect size distribution, if a design rule is relaxed, for instance, if the line width and line spacing are widened, the total number of yield defects decreases and the critical area increases. From the limitation of applications to interconnect substrates, the critical area can be obtained by the union of the critical area for line shorts and the critical area for line opens. Then, the Poisson yield is expressed as a function of line width, and trade-offs of the design rule change can be studied. If an increase in the design rule has minimal impact on the overall substrate area, then yield improvement by increasing the design rules can lead to a more cost-effective substrate.

Cunningham et al. [9.44] presented a common-yield model to analyze and compare the yield of products from different facilities using a linear regression model. *Berglund* [9.45] developed a variable defect size yield model. *Milchalka* [9.17] presented a yield model that considers the repair capability in a part of the die area. *Stapper* and *Rosner* [9.37] presented a yield model using the number of circuits and average number of yield defects per circuit. *Dance* and *Jarvis* [9.46] explained the application of yield models to accelerate yield learning and to develop a performance–price improvement strategy.

Choosing a yield model is basically an experiential process. IC manufacturers compare data from a specific process for yield versus die size using various models and select the best fit. Depending on the distribution of die sizes of a given product and the distribution pattern of the defects, different yield models will best fit the data [9.47].

9.1.4 Issues for Yield Improvement

Achieving high-yield devices is a very challenging task due to reduced process margins and increased IC design complexity. Recent research has emphasized the role of parametric yield loss as well as that of functional yield loss in proactive yield management. Although random yield loss typically dominates in high-volume production, systematic and parametric yield losses become more important when a fabrication process is newly defined and is being tuned to achieve the necessary processes and device parameters [9.48]. Considerable attention has been paid thus far to improving random yield, but relatively little attention has been paid to systematic and parametric yield problems. With new technologies, a process may never be stabilized and statistical device-parameter variations will be a big headache. Traditionally, parametric yield problems were addressed after a design was manufactured. Low-yielding wafers were investigated to identify what process variations caused the yield loss. Then, simulations were used to see where the design should be changed to improve the yield. The traditional redesign approach is very costly compared to handling design at the front-end of the design process using design for yield (DFY) techniques. The use of DFY techniques accelerates the design flow, reduces cycle times and provides higher yield.

Before a high-volume chip comes to market, it must be manufacturable at an acceptable yield. Although traditionally yield issues have been in the domain of manufacturing teams, a new approach to bridge the gap between design and manufacture is necessary as chip geometry shrinks. *Peters* [9.49] emphasized the increasing role of DFY approaches in leading-edge device manufacturability to allow for tuning of all test programs and models so that design, manufacturing and testing provide high-yielding devices.

Li et al. [9.48] presented a holistic yield-improvement methodology that integrates process recipe and design information with in-line manufacturing data to solve the process and design architecture issues that

affect yield and performance. The approach suggests improving yield not just by eliminating defects but also by resolving parametric problems.

Nardi and *Sangiovanni-Vincentelli* [9.12] observed that for complex nanodesigns functional yield might depend more on the design attributes than on the total chip area. Given that the current yield-aware flow optimizes yield at the layout level after optimizing speed and area, a synthesis-for-manufacturability approach is suggested in which manufacturability replaces area in the cost function.

Segal [9.38] claimed that each new technology generation will see lower and lower yields if the defect level of well-running processes is not reduced. A strategy to reduce the defect level is to encompass techniques for responding quickly to defect excursion using in-line wafer scanners and wafer position tracking. The defect excursion strategy eliminates wafers and lots with very high defect densities.

9.2 Semiconductor Reliability

Once an IC device is released to the user, an important and standard measure of device performance is reliability, which is defined as the probability of a device conforming to its specifications over a specified period of time under specified conditions. A failure rate function is usually used to describe device reliability, which is defined for a population of nonrepairable devices as the instantaneous rate of failure for the surviving devices during the next instant of time. If $h(x)$ denotes a failure rate function, the corresponding reliability function is expressed by

$$R(t) = \mathrm{e}^{-\int_0^t h(x)\,\mathrm{d}x} .$$

In this section, the failure rate in semiconductor device reliability is explained. Then, we discuss where each semiconductor failure mechanism occurs in the bathtub failure rate. Finally, techniques used for reliability improvement are reviewed.

9.2.1 Bathtub Failure Rate

When engineers have calculated the failure rate of a semiconductor population over many years, they have commonly observed that the failure rate is described by a bathtub shape.

Initially, semiconductor devices show a high failure rate, resulting in an infant mortality period. The infant mortality period results from weak devices that have shorter lifetimes than the normal stronger devices, implying that infant mortality period applies to a whole population rather than a single device. The operating period that follows the infant mortality period has a lower, and almost constant, failure rate and is called the useful life period. Infant mortality and useful life failures are due to defects introduced during the manufacturing process, such as particle defects, etch defects, scratches and package assembly defects.

A device that has reached the end of its useful life enters the final phase called the aging period. Failures during the aging period are typically due to aging or cumulative damage, and these can be avoided by careful technology development and product design. These failures are inherent process limitations and are generally well-characterized.

The semiconductor manufacturing process requires hundreds of sequential steps and thus hundreds, or even thousands, of process variables must be strictly controlled to maintain the device reliability. Despite the exponential scaling of semiconductor size and chip complexity, IC reliability has increased at an even faster rate as reliability engineers reduce infant mortality and useful life failure rate and push the aging period beyond the typical usage period through a variety of reliability improvement techniques.

9.2.2 Occurrence of Failure Mechanisms in the Bathtub Failure Rate

Failure mechanisms of semiconductor devices can be classified into three groups: electrical stress failures, intrinsic failures and extrinsic failures [9.7, 50].

Electrical stress failures are user-related, and the major causes are electrical-over-stress (EOS) and electrostatic discharge (ESD) due to improper handling. ESD and EOS problems are thoroughly discussed in *Vinson* and *Liou* [9.51]. Because this failure mechanism is event-related, it can occur anywhere in the infant mortality period, the useful life period or the aging period.

The intrinsic failure mechanism results from all crystal-related defects, and thus it occurs predominantly in the infant mortality period but rarely in the aging period.

On the other hand, extrinsic failures are the result of device packaging, metallization and radiation and they can occur any time over the device's lifetime. Extrinsic failures that are due to process deficiencies, such as migration and microcracks, occur during the infant mortality period. The extrinsic failure mechanisms related to packaging deficiency, such as bond looping and metal degradation, occur in the aging period. The radiation-related extrinsic failure mechanisms, such as bit flips due to external radiation, occur continuously over the device lifetime [9.50].

The terms extrinsic and intrinsic failure have also been used in different contexts. First, intrinsic failure is used to describe those failures that are due to internal causes of a device, while failures due to forces external to the product, such as mishandling or accidents, are called extrinsic failures [9.52]. In this case, intrinsic failures occur in the infant mortality period or in the aging period, while extrinsic failures occur in the useful life period. Secondly, the terms intrinsic and extrinsic failure are used to classify oxide failures [9.53]. In this case, intrinsic failures are due to the breakdown of oxide which is free of manufacturing defects, and thus is usually caused by an inherent imperfection in the dielectric material. These failures occur in the aging period at an increasing failure rate. On the other hand, extrinsic failures that result from process defects in the oxide or problems in the oxide fabrication occur in the infant mortality period.

9.2.3 Issues for Reliability Improvement

As reliability engineers have recognized that it is no longer affordable to handle reliability assurance as a back-end process in IC product development, the reliability emphasis has been shifted from end-of-line statistical-based stress testing to new proactive techniques such as design for reliability (DFR), built-in reliability (BIR), wafer-level reliability (WLR), qualified manufacturing line (QML), and physics-of-failure (POF) approaches [9.54, 55].

DFR means building reliability into the design rather than incorporating it after development [9.56]. The importance of DFR increases as stress testing becomes increasingly difficult as the allowable stress is decreased.

The effectiveness of BIR has been outlined in [9.57, 58] for manufacturing highly reliable ICs through the elimination of all possible defects in the design stage.

WLR represents a transition from the end-of-line concept toward the concept of BIR, because the testing is performed at the wafer level reducing the time and expense of packaging. Examples of WLR implementation into a production line or a testing method are given in [9.59–62].

QML is another evolutionary step devised for the purpose of developing new technologies where the manufacturing line is characterized by running test circuits and standard circuit types [9.63]. Understanding failure mechanisms and performing failure analysis are critical elements in implementing the BIR and QML concept.

In cases where the fundamental mechanical, electrical, chemical, and thermal mechanisms related to failures are known, it is possible to prevent failures in new products before they occur. This is the basic idea of POF, which is the process of focusing on the root causes of failure during product design and development in order to provide timely feedback.

9.3 Burn-In

Burn-in is a production process that operates devices, often under accelerated environments, so as to detect and remove weak devices containing manufacturing defects before they are sold or incorporated into assemblies. Because the design rules change so quickly, burn-in today is an essential part of the assembly and testing of virtually all semiconductor devices. To burn-in or not to burn-in and how long the burn-in should be continued are perennial questions.

In this section, we discuss several issues related to burn-in, such as key questions for burn-in effectiveness, burn-in level and burn-in types. Then, the previous burn-in literature is reviewed based on the level of burn-in application.

9.3.1 The Need for Burn-In

Since most semiconductor devices ordinarily have an infant mortality period, the reliability problem during this period becomes extremely important. Manufacturers use burn-in tests to remove infant mortality failures for most circuits, especially where high reliability is a must. Burn-in ensures that a circuit at assembly has moved to the

useful life period of the bathtub curve. During burn-in, elevated voltage and temperature are often combined to activate the voltage- and temperature-dependent failure mechanisms for a particular device in a short time. Careful attention to design of stress burn-in is necessary to ensure that the defect mechanism responsible for infant mortality failures is accelerated while normal strong devices remain unaffected.

Although burn-in is beneficial for screening in the infant mortality period, the burn-in cost ranges from 5–40% of the total device cost depending on the burn-in time, quantities of ICs and device complexity [9.64], and it might introduce additional failures due to EOS, ESD or handling problems. Solutions to the key questions posed by *Kuo* and *Kuo* [9.65] will continue to be found with new technologies for exercising burn-in effectively:

1. How much should infant mortality be reduced by burn-in?
2. Under what environmental conditions should burn-in be performed?
3. Should burn-in be accomplished at the system, subsystem, or component level?
4. Who should be in charge of burn-in, the vender, the buyer, or a third party?
5. Are there any side-effects of burn-in?
6. How will the industry benefit from burn-in data?
7. What physics laws should be followed to conduct burn-in?

9.3.2 Levels of Burn-In

There are three burn-in types based on levels of a device: package-level burn-in (PLBI), die-level burn-in (DLBI), and wafer-level burn-in (WLBI) [9.66–68].

PLBI is the conventional burn-in technology where dies are packed into the final packages and then subjected to burn-in. Although PLBI has the advantage of assuring the reliability of the final product, repairing or discarding a product after PLBI is far too costly.

The strong demand for known good dies (KGD) has motivated the development of more efficient burn-in technology. Generally, KGD is defined as a bare unpacked die that has been tested and verified as fully functional to meet the full range of device specifications at a certain level of reliability [9.68, 69]. KGD enables manufacturers to guarantee a given quality and reliability level per die before integration and assembly. Optimizing burn-in is a key aspect of KGD [9.69].

In DLBI, dies are placed in temporary carriers before being packed into their final form to reduce the cost of added packaging. DLBI and testing of the individual die before packaging ensures that only KGD are packaged and thus produces a quality product at a reduced cost.

Considerations of how to reduce burn-in cost and solve KGD issues have led to the concept of WLBI. WLBI achieves burn-in on the wafer as soon as it leaves the fab. Though WLBI can result in less-reliable final products than PLBI, the trend in industry is to do more testing at the wafer level due to the cost and KGD issues [9.70].

Recently, the line between burn-in and testing has begun to blur as far as reducing testing costs and cycle times. For example, some test functions have moved to the burn-in stage and multi-temperature environments have moved to final testing. DLBI and WLBI that have evolved from burn-in to include testing are called die-level burn-in and testing (DLBT) and wafer-level burn-in and testing (WLBT), respectively. It is reported that DLBT is an expensive step in memory production and the transfer to WLBT can reduce the overall back-end cost by 50% [9.71].

9.3.3 Types of Burn-In

A basic burn-in system includes burn-in sockets to provide a temporary electrical connection between the burn-in board (BIB) and the device under test (DUT) package. Each BIB might accommodate 50 or more sockets, and a burn-in system might hold 32 BIBs. To develop a successful burn-in strategy, detailed knowledge is necessary about temperature distributions across a DUT package, across a BIB, and throughout the burn-in oven [9.72].

Three burn-in types are known to be effective for semiconductor devices: steady-state or static burn-in (SBI), dynamic burn-in (DBI) and test during burn-in (TDBI) [9.7, 73].

In SBI, DUTs are loaded into the burn-in boards (BIB) sockets, the BIBs are put in the burn-in ovens and the burn-in system applies power and an elevated temperature condition (125–150 °C) to the devices for a period ranging from 12 to 24 h. Once the devices cool down, the BIBs are extracted from the boards. These devices are placed in handling tubes and mounted on a single-device tester. Functional tests are then applied on the devices to sort them according to failure types. Because the DUT is powered but not exercised electrically, SBI may not be useful for complex devices because external biases and loads may not stress internal nodes.

In DBI, the DUT is stimulated at a maximum rate determined by the burn-in oven electronics, which can

propagate to internal nodes. Neither SBI not DBI monitors the DUT response during the stress, and thus dies that fail burn-in cannot be detected until a subsequent functional test.

Beyond static and dynamic burn-in is so-called intelligent burn-in [9.72]. Intelligent burn-in systems not only apply power and signals to DUTs, they also monitor DUT outputs. Therefore, they can guarantee that devices undergoing burn-in are indeed powered up and that input test vectors are being applied. In addition, they can perform some test functions. TDBI is a technique for applying test vectors to devices while they are being subjected to stresses as part of the burn-in process. Though function testing is not possible due to the burn-in stress, idle time can be used advantageously to verify circuit integrity, permitting abbreviated functional testing after burn-in.

9.3.4 Review of Optimal Burn-In Literature

While a considerable number of papers have dealt with burn-in at one level, recent research has been directed to the study of burn-in at multiple levels. In this section, we will review the burn-in literature based on the burn-in level being analyzed.

One-Level Burn-In

To fix burn-in at one level, previous work has taken two different approaches: the black-box approach and the white-box approach. In the black-box approach, each device is treated as a black box and a specific failure rate distribution is assumed for the device. In the white-box approach, the device is decomposed into smaller components and a failure rate distribution is assumed for each component. Then the whole-device failure rate is obtained from the structure function and component failure rate.

Many papers have taken the black-box approach and determined the optimal burn-in time to minimize a cost function. *Mi* [9.74, 75] showed that optimal burn-in times that minimize various cost functions occur in the infant mortality period. *Sheu* and *Chien* [9.76] showed the same result for two different types of failures. Assuming that the device lifetime follows a Weibull distribution, *Drapella* and *Kosznik* [9.77] obtained optimal burn-in and preventive replacement periods using Mathcad code. *Cha* [9.78–80] considered a minimally repaired device and derived the properties of optimal burn-in time and block replacement policy. *Tseng* and *Tang* [9.81] developed a decision rule for classifying a component as strong or weak and an economical model to determine burn-in parameters based on a Wiener process. Assuming a mixed Weibull distribution, *Kim* [9.82] determined optimal burn-in time with multiple objectives of minimizing cost and maximizing reliability. A nonparametric approach [9.83] and a nonparametric Bayesian approach [9.84] have been used to estimate the optimal system burn-in time that minimizes a cost function.

The first report that takes a white-box approach appears in *Kuo* [9.85]. The optimal component burn-in time was determined to minimize a cost function subject to a reliability constraint, assuming that the failure of each component follows a Weibull distribution. *Chi* and *Kuo* [9.86] extended it to include a burn-in capacity constraint. *Kar* and *Nachlas* [9.87] consider a series structure, assuming that each component has a Weibull distribution. Given that each component that fails system burn-in is replaced, the optimal system burn-in time was determined to maximize a net-profit function that balances revenue and cost. For the case where percentile residual life is the performance measure of burn-in, *Kim* and *Kuo* [9.88] studied the relationship between burn-in and percentile residual life.

Multi-level Burn-in

For studying burn-in at various levels, the white-box approach must be asked to characterize the failure time distribution of the whole device. Because system burn-in is never necessary after component burn-in if assembly is perfect [9.89, 90], modeling of burn-in at multiple levels must focus on the quantification of assembly quality. *Whitbeck* and *Leemis* [9.91] added a pseudo-component in series to model the degradation of a parallel system during assembly. Their simulation result showed that system burn-in is necessary after component burn-in to maximize the mean residual life. *Reddy* and *Dietrich* [9.92] added several connections to explain an assembly process and assumed that each of components and connections followed a mixed exponential distribution. The optimal burn-in time at the component and system levels were determined numerically to minimize the cost functions, given that the components were replaced and the connections minimally repaired upon failure. *Pohl* and *Dietrich* [9.93] considered the same problem for mixed Weibull distributions. *Kuo* [9.94] used the term *incompatibility* for reliability reduction realized during assembly process. The incompatibility factor exists not only at the component level but also at the subsystem and the system levels due to poor manufacturability, workmanship, and design strategy. *Chien* and *Kuo* [9.95] proposed a nonlinear model

to estimate the optimal burn-in times for all levels as well as to determine the number of redundancies in each subsystem when incompatibility exists. To quantify the incompatibility factor, *Chien* and *Kuo* [9.96] added a uniform random variable to the reliability function. Optimal burn-in times at different levels were determined to maximize the system reliability, subject to a cost constraint via simulation, assuming that the component followed a Weibull distribution. A conceptual model has been developed [9.97] that considers PLBI and WLBI for minimizing a cost function subject to the reliability requirement. *Kim* and *Kuo* [9.98, 99] analytically derived the conditions for system burn-in to be performed after component burn-in using a general system distribution to which the component burn-in information and assembly problems were transferred. *Kim* and *Kuo* [9.100] presented another model for quantifying the incompatibility factor when the assembly adversely affected the components that were replaced at failure. Optimal component and system burn-in times were determined using nonlinear programming for various criteria.

9.4 Relationships Between Yield, Burn-In and Reliability

As semiconductor technology advances, burn-in is becoming more expensive, time-consuming and less capable of identifying the failure causes. Previous research has focused on the determination of burn-in time based on a reliability function estimated from the time-to-first-failure distribution. However, newer proactive methods to determine the burn-in period in the early production stage are of great interest to the semiconductor industry.

One such approach is based on the relation model of yield, burn-in and reliability, which we will review in this section.

9.4.1 Background

Observing that high yield tends to go with high reliability, it was conjectured that defects created on IC devices during manufacturing processes determine yield as well as reliability [9.31]. Subsequent experiments confirmed that each defect in a device affects either yield or reliability, depending on its size and location. This is illustrated in Fig. 9.2 for oxide defects. Therefore, reliability can be estimated based on yield if the relationship between yield and reliability is identified. A model that relates yield and reliability has many applications, such as in yield and reliability predictions for future devices, device architecture design, process control and specification of allowable defect density in new processes for achieving future yield and reliability goals. As a result, the start-up time of new fabrication facilities and cycle times can be shortened by reducing the amount of traditional stress testing required to qualify new processes and products.

Developing a relation model of yield and reliability has been an active research area in the past decade. Three different definitions have been used for reliability in previous research. First, reliability is defined by the probability of a device having no reliability defects, where a reliability defect is defined not as a function of the operating time but as a fixed defect size. Secondly,

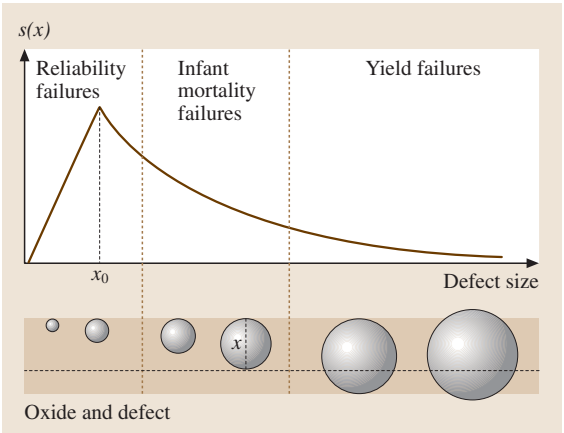


Fig. 9.2 Defect size distribution and oxide problems [9.15]

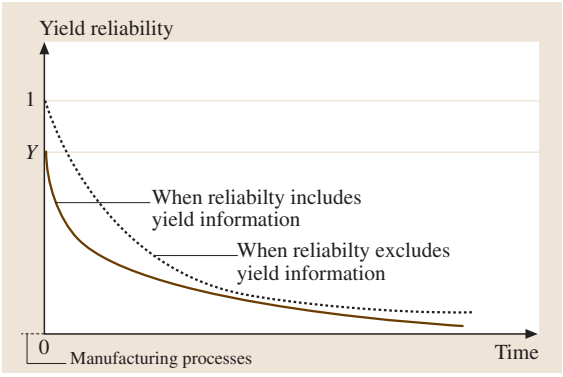


Fig. 9.3 Yield-reliability relationship depending on the definition of reliability [9.24]

reliability denotes the probability of a device having no reliability defects given that there are no yield defects. This reliability is equivalent to yield at time zero, as depicted in Fig. 9.3. This definition of reliability is useful from the designer's point-of-view as it incorporates yield information, but it is not consistent with the traditional definition. Thirdly, reliability is defined as the probability of a device having no reliability defects by time t. In this case, reliability is defined as a function of the operating time without incorporating the yield defects, after assuming that any device that is released to field operation has passed a manufacturing yield test, implying that no yield defects exist. Such a reliability is always 1 at time zero.

9.4.2 Time-Independent Reliability without Yield Information

The first model that related yield and reliability was reported by *Huston* and *Clarke* [9.23]. Let N_y be the number of yield defects and N_r be the number of reliability defects in a device. Reliability defects were defined as a specific defect size rather than as a function of time. Assume that N_y and N_r are independent and each follows a Poisson distribution. Thus, the distribution of N_y is given in (9.1) and the distribution of N_r is given by

$$P^*_{\text{Poisson}}(k) = P(N_r = k) = \frac{e^{-\lambda_r}\lambda_r^k}{k!}, \quad k = 0, 1, 2,$$

where λ_r is the average number of reliability defects per chip. Then, for a device without reliability defects, the Poisson reliability model is obtained by

$$R_{\text{Poisson}} = P^*_{\text{Poisson}}(0) = e^{-\lambda_r}. \tag{9.7}$$

The Poisson yield–reliability relation is obtained from (9.2) and (9.7) by

$$R_{\text{Poisson}} = Y^{\gamma}_{\text{Poisson}} \tag{9.8}$$

where

$$\gamma = \frac{\lambda_r}{\lambda_y}. \tag{9.9}$$

Next, they expressed $\lambda_y = A_y D_0$ and $\lambda_r = A_r D_0$ where D_0 is the common defect density for the yield and reliability defects, and A_y and A_r are the yield and reliability critical areas, respectively.

Subsequently, *Kuper* et al. [9.101] used a similar model given by

$$R_{\text{Poisson}} = (Y_{\text{Poisson}}/M)^{\gamma} \tag{9.10}$$

where M is the maximum possible yield fraction considering clustering effects and edge exclusions. The value of γ depends on the technology and process and on the conditions under which the product is used. They assumed that $\lambda_y = AD_y$ and $\lambda_r = AD_r$ where A is the device area and D_y and D_r are the yield and reliability defect density, respectively. The model was verified with high-volume ICs manufactured by several processes.

Riordan et al. [9.27] verified that (9.10) agrees well for yields based on the lot, the wafer, the region of the wafer and the die in a one-million-unit sample of microprocessors. *Van der Pol* et al. [9.102] used (9.10) to study the IC yield and reliability relationship further for 50 million high-volume products in bipolar CMOS and BICMOS technologies from different wafer fabrication facilities. Experiments showed that a clear correlation exists among functional yield, burn-in failures and field failures.

Zhao et al. [9.103] used a discrete exponential yield model given in (9.6) for yield and (9.7) for reliability. Then, the relation model is obtained by

$$R = \exp\left(-\frac{m\left(1 - Y^{1/m}_{\text{discrete expo}}\right)}{Y^{1/m}_{\text{discrete expo}}}\gamma\right).$$

9.4.3 Time-Independent Reliability with Yield Information

Barnett et al. [9.19] developed a relation model for the negative binomial model, rather than for the Poisson model, assuming that the number of reliability defects is proportional to the number of yield defects in a device. Let N be the total number of defects, where $N = N_y + N_r$. Then

$$P(N_y = m, N_r = n | N = q) = \binom{q}{m} p_y^m p_r^n, \tag{9.11}$$

where p_y is the probability of a defect being a yield defect, and $p_r = 1 - p_y$) is the probability of a defect being a reliability defect. Let $\lambda = E(N)$. If N is assumed to follow a negative binomial distribution

$$P(N = q) = \frac{\Gamma(\alpha + q)}{q!\Gamma(\alpha)} \frac{\left(\frac{\lambda}{\alpha}\right)^q}{(1 + \frac{\lambda}{\alpha})^{\alpha+q}},$$

then the wafer probe yield can be obtained by

$$Y_{\text{nb}} = P(N_y = 0) = \left(1 + \frac{\lambda_y}{\alpha}\right)^{-\alpha}, \tag{9.12}$$

where $\lambda_y = \lambda p_y$ is the average number of yield defects. Let R be the conditional probability that there are no

reliability defects given that there are no yield defects. Then,

$$R = P(N_r = 0 | N_y = 0) = \left(1 + \frac{\lambda_r(0)}{\alpha}\right)^{-\alpha} , \quad (9.13)$$

where

$$\lambda_r(0) = \frac{\lambda p_r}{1 + \lambda p_y / \alpha}$$

is the average number of reliability defects given that there are no yield defects. Using (9.12) and (9.13), the relation model is derived as

$$R = \left[1 + \gamma \left(1 - Y^{1/\alpha}\right)\right]^{-\alpha} ,$$

where $\gamma = \lambda_r / \lambda_y = p_r / p_y$. Numerical examples were used to show that the number of reliability failures predicted by the negative binomial model can differ from the prediction by the Poisson model because of clustering effects. *Barnett* et al. [9.18] modified the model in order to consider the possibility of repair in a certain area of a chip and experimentally verified that the reliability of an IC with a given number of repairs can be accurately quantified with the model. *Barnett* et al. [9.21] and [9.20] validated the yield–reliability relation model using yield and stress test data from a 36-Mbit static random-access memory (SRAM) memory chip and an 8-Mbit embedded dynamic random-access memory (DRAM) chip and from 77 000 microprocessor units manufactured by IBM microelectronics, respectively.

9.4.4 Time-Dependent Reliability

Van der Pol et al. [9.104] added the time aspect of reliability to their previous model [9.102] to suggest detailed burn-in. From an experiment, a combination of two Weibull distributions was employed for the time-to-failure distribution by which $1 - R_{\text{Poisson}}$ in (9.8) is replaced. Similarly, *Barnett* and *Singh* [9.22] introduced the time aspect of reliability in (9.13) using a Weibull distribution. *Forbes* and *Arguello* [9.105] expressed the reliability by time t by

$$\begin{aligned} R(t) &= 1 - e^{-\lambda_r(t)} \simeq 1 - \lambda_r(t) \\ &= 1 - A D_r(t) = 1 - A D_y \gamma(t) , \end{aligned} \quad (9.14)$$

where $\gamma(t) = \frac{D_r(t)}{D_y}$. Then, the Weibull distribution reliability replaces the left-hand side of (9.14) and the corresponding relationship of yield and reliability is used to optimize the burn-in period. All of these models are based on the assumption that the device time-dependent reliability is available in advance from experiments or field failure data.

Kim and *Kuo* [9.26] suggested using $\lambda_r(t) = A_r(t) D_0$ in (9.8), where $\lambda_r(t)$ denotes the mean number of reliability defects realized by time t, and $A_r(t)$ is the reliability critical area by time t. Assuming that the defect growth for operation time t is a known increasing function of time, they calculated $\lambda_r(t)$ and derived a relation model of oxide yield and time-dependent reliability. This is the first model in which time-dependent reliability is estimated from yield and critical area analysis, rather than from field failure data. Because of the properties of the assumed defect growth function, the resulting reliability function has an increasing failure rate. The effect of burn-in on yield, using yield and reliability critical area, was studied by *Kim* et al. [9.106]. *Kim* et al. [9.24] presented another model to tie oxide yield to time-dependent reliability by combining the oxide time to a breakdown model with the defect size distribution given in (9.4). This reliability model predicted from the yield has an infant mortality period such that the optimal burn-in policy for burn-in temperature, burn-in voltage and burn-in time can be determined based on the model.

To handle the dependence between the numbers of yield and reliability defects, *Kim* and *Kuo* [9.25, 107] used a multinomial distribution for the number of yield defects, the number of reliability defects that fail during burn-in and the number of reliability defects that are eventually released to field operation. The distribution of the number of defects is arbitrary. From a feature of multinomial distribution, the number of yield defects

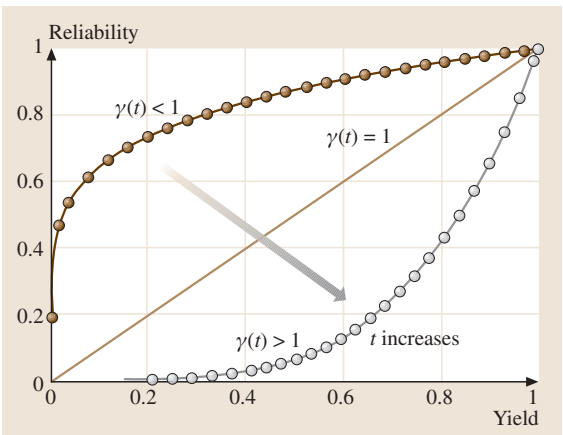


Fig. 9.4 Relation between yield and time-dependent reliability [9.25]

and the number of reliability defects are negatively correlated if the total number of defects in a device is fixed. An analytical result showed that two events, the number of yield defects being zero and the number of reliability defects that fail during burn-in being zero, are positively correlated. This explains the correlated improvement between yield and burn-in fallout. It was also shown that burn-in may be useful if device-to-device variability in the number of defects passing yield tests is greater than a threshold, where the threshold depends on the failure rate of a defect occurrence distribution and the number of defects remaining after the test. Let $\gamma(t)$ be the scaling factor from yield to reliability such that

$$\gamma(t) = \frac{\lambda_r(t)}{\lambda_y} ,$$

where $\lambda_r(t)$ is the number of reliability defects failed by time t. Figure 9.4 shows that a larger value of the scaling factor gives a smaller value of reliability for a given yield value. Clearly, burn-in, reliability and warranty cost can be controlled in an actual process by considering yield and the scaling factor. One can conjecture that burn-in should be performed if the scaling factor is large.

9.5 Conclusions and Future Research

In this chapter, we reviewed semiconductor yield, burn-in and reliability modeling and analysis as a fundamental means of proactive yield and reliability management. It was emphasized that with new technologies the consideration of parametric and systematic yield loss is increasingly important in addition to the consideration of yield defects. Therefore, developing a robust design methodology that can be used to improve parametric and systematic yield becomes a promising research area. Statistical softwares for easily implementing the response surface methodology and Monte Carlo simulation are necessary to overcome the limitations of the current corner analysis method that is widely used in parametric yield analysis.

As design rules tend to change quickly, whether or not to perform burn-in is a perennial question. Previously, a considerable number of papers have studied ways to determine optimal burn-in times based on time-to-first-failure distributions, such as the Weibull distribution or the mixed Weibull distribution. Since burn-in is expensive and time-consuming, more proactive approaches are necessary for determining optimal burn-in time, for example POF analysis.

As correlated improvements in yield, burn-in failures and reliability have occurred, the development of a model relating them has been an active research area in the last decade. Such a model is a prerequisite to predict and control burn-in and reliability based on the device layout in the design stage. Through the model, cycle times and testing costs can be reduced significantly. Currently, experiments are validating the relationship between yield and time-independent reliability. Experiments are necessary to confirm the time-dependent relationship as well. Validation of the time-dependent behavior of reliability defects using IC devices is necessary to determine optimal burn-in periods through the relation model. To do this, physical models must be available to characterize defect growth during operation for various device types, which will enable the estimation of reliability defects as a function of operation time. Also, some future research should be conducted to generalize the yield–reliability relation model to other defect density distributions besides the Poisson and negative binomial models.

References

9.1 Semiconductor Industry Association: *The National Technology Roadmap for Semiconductors* (Semiconductor Industry, San Jose, CA 2003)

9.2 World Semiconductor Trade Statistics: *Semiconductor Market Forecast* (Miyazaki, Japan May 2004)

9.3 W. T. K. Chien, W. Kuo: Use of the Dirichlet process for reliability analysis, Comput. Ind. Eng. **27**, 339–343 (1994)

9.4 W. T. K. Chien, W. Kuo: Extensions of Kaplan–Meier estimator, Commun. Stat. Simul. Comp. **24**, 953–964 (1995)

9.5 National Research Council Committee: *Implications of Emerging Micro- and Nano-technologies*, National Research Council Committee, Division of Engineering and Physical Sciences (National Academy, Washington 2003) p. 251

9.6 W. Luo, Y. Kuo, W. Kuo: Dielectric relaxation and breakdown detection of doped tantalum oxide high-k thin films, IEEE Trans. Dev. Mater. Reliab. **4**, 488–494 (2004)

9.7 W. Kuo, W. T. K. Chien, T. Kim: *Reliability, Yield, and Stress Burn-in* (Kluwer Academic, Norwell 1998)

9.8 C. J. McDonald: New tools for yield improvement in integrated circuit manufacturing: can they be applied to reliability, Microelectron. Reliab. **39**, 731–739 (1999)

9.9 L. Peters: Introduction to Integrated Yield Analysis, Semicond. Int. **22**, 56 (January 1999)

9.10 http://www.cadence.com/whitepapers/5047ParametricYieldWPfnl.pdf

9.11 R. K. Ravikumar: Viewpoint addressing design yield prior to silicon, Chip Des. Mag. **40** (April/May 2004)

9.12 A. M. Nardi, A. Sangiovanni-Vincentelli: Logic synthesis for manufacturability, IEEE Des. Test Comput. **21**(3), 192–199 (2004)

9.13 A. J. Strojwas: *Yield of VLSI Circuits: myths vs. reality*, 23th Design Automation Conference (IEEE Computer Society, Las Vegas, NV 1986) pp. 234–235

9.14 C. H. Stapper: Large area fault clusters and fault tolerance in VLSI circuits: a review, IBM J. Res. Develop. **33**, 174–177 (1989)

9.15 W. Kuo, T. H. Kim: An overview of manufacturing yield and reliability modeling for semiconductor products, Proc. IEEE **87**, 1329–1344 (1999)

9.16 A. V. Ferris-Prabhu: *Introduction to Semiconductor Device Yield Modeling* (Artech House, Boston 1992)

9.17 T. L. Michalka, R. C. Varshney, J. D. Meindl: A discussion of yield modeling with defect clustering, circuit repair and circuit redundancy, IEEE Trans. Semicond. Manuf. **3**, 116–127 (1990)

9.18 T. S. Barnett, A. D. Singh, V. P. Nelson: *Estimating burn-in fall-out for redundacy memory*, Proc. 2001 VLSI Test Symp. (IEEE Computer Society, Los Angeles, CA May 2001) pp. 340–347

9.19 T. S. Barnett, A. D. Singh, V. P. Nelson: Extending integrated-circuit yield models to estimate early life reliability, IEEE Trans. Reliab. **52**, 296–301 (2003)

9.20 T. S. Barnett, A. D. Singh, M. Grady, V. P. Nelson: *Yield-reliability modeling: experimental verification and application to burn-in reduction*, Proc. 20th IEEE VLSI Test Symp. (IEEE Computer Society, Monterey, CA 2002) pp. 1–6

9.21 T. S. Barnett, M. Grady, K. Purdy, A. D. Singh: Redundancy implications for early-life reliability: experimental verification of an integrated yield–reliability model. In: *ITC International Test Conference* (IEEE Computer Society, Baltimore, MD 2002) pp. 693–699

9.22 T. S. Barnett, A. D. Singh: *Relating yield models to burn-in fall-out in time*, ITC Int. Test Conf. (IEEE Computer Society, Charlotte, NC 2003) pp. 77–84

9.23 H. H. Huston, C. P. Clarke: *Reliability defect detection and screening during processing: theory and implementation*, Proc. Int. Reliability Physics Symposium (IEEE Reliability Society, San Diego, CA 1992) pp. 268–275

9.24 K. O. Kim, W. Kuo, W. Luo: A relation model of gate oxide yield and reliability, Microelectron. Reliab. **44**, 425–434 (2004)

9.25 K. O. Kim, W. Kuo: A unified model incorporating yield, burn-in and reliability, Naval Res. Log. **51**, 704–719 (2004)

9.26 T. H. Kim, W. Kuo: Modeling manufacturing yield and reliability, IEEE Trans. Semicond. Manuf. **12**, 485–492 (1999)

9.27 W. C. Riordan, R. Miller, J. Hicks: Reliability versus yield and die location in advanced VLSI, Microelectron. Reliab. **39**, 741–749 (1999)

9.28 J. A. Cunningham: The use and evaluation of yield models in integrated circuit manufacturing, IEEE Trans. Semicond. Manuf. **3**, 60–71 (1990)

9.29 M. Raghavachari, A. Srinivasan, P. Sullo: Poisson mixture yield models for integrated circuits: a critical review, Microelectron. Reliab. **37**, 565–580 (1997)

9.30 R. E. Langford, J. J. Liou: *Negative binomial yield model parameter extraction using wafer probe bin map data* (IEEE Electron Devices Meeting, Hong Kong 1998) pp. 130–133

9.31 F. Jensen: *Yield, quality and reliability – a natural correlation?*, Reliability 1991, ed. by R. H. Matthews (Elsevier Applied Science, London 1993) pp. 739–750

9.32 Z. Stamenkovic, N. Stojadinovic: New defect size distribution function for estimation of chip critical area in integrated circuit yield models, Electron. Lett. **28**, 528–530 (1992)

9.33 Z. Stamenkovic, N. Stojadinovic: Chip yield modeling related to photolithographic defects, Microelectron. Reliab. **32**, 663–668 (1992)

9.34 G. Allen, A. Walton, R. Holwill: A yield improvement technique for IC layout using local design rules, IEEE Trans. CAD ICs Syst. **11**, 1355–1362 (1992)

9.35 S. Williamson, A. Singh, V. Nelson: *Fault and yield modeling of MCMs for automotive applications*, Proc. Int. MCM Conf. (IEEE Society, Denver 1996) pp. 147–154

9.36 J. Mattick, R. Kelsall, R. Miles: Improved critical area prediction by application of pattern recognition techniques, Microelectron. Reliab. **36**, 1815–1818 (1996)

9.37 C. H. Stapper, R. J. Rosner: Integrated circuit yield management and yield analysis: development and implementation, IEEE Trans. **8**, 95–102 (1995)

9.38 J. Segal, L. Milor, Y. K. Peng: Reducing baseline defect density through modeling random defect-limited yield, Micro Mag. **18**, 61–71 (2000)

9.39 H. Sato, M. Ikota, A. Sugimoto, H. Masuda: A new defect distribution metrology with a consistent discrete exponential formula and its application, IEEE Trans. Semicond. Manuf. **12**, 409–418 (1999)

9.40 K. S. Park, C. H. Jun: Semiconductor yield models using contagious distributions and their limiting forms, Comput. Eng. **42**, 115–125 (2002)

9.41 C. H. Jun, Y. Hong, Y. K. Kim, K. S. Park, H. Park: A simulation-based semiconductor chip yield model incorporating a new defect cluster index, Microelectron. Reliab. **39**, 451–456 (1999)

9.42 J. A. Carrasco, V. Suñé: Combinatorial methods for the evaluation of yield and operational reliability of fault-tolerant systems-on-chip, Microelectron. Reliab. **44**, 339–350 (2004)

9.43 M. Scheffler, D. Cottet, G. Tröster: A simplified yield modeling method for design rule trade-off in interconnection substrates, Microelectron. Reliab. **41**, 861–869 (2001)

9.44 S. P. Cunningham, C. J. Spanos, K. Voros: Semiconductor yield improvement: results and best practices, IEEE Trans. Semicond. Manuf. **8**, 103–109 (1995)

9.45 C. N. Berglund: A unified yield model incorporating both defect and parametric effects, IEEE Trans. Semicond. Manuf. **9**, 447–454 (1996)

9.46 D. Dance, R. Jarvis: Using yield models to accelerate learning curve progress, IEEE Trans. Semi. Manuf. **5**, 41–45 (1992)

9.47 L. Peters: Choosing the Best Yield Model for Your Product, Semicond. Int. **23**, 52 (May 2000)

9.48 X. Li, J. Strojwas, M. F. Antonelli: Holistic Yield Improvement Methodology, Semiconductor Fabtech **27**(7), 257–265 (1998)

9.49 L. Peters: Demystifying Design-for-Yield, Semicond. Int. **27**, 42 (July 2004)

9.50 A. Amerasekera, D. S. Campbell: *Failure Mechanisms in Semiconductor Devices* (Wiley, New York 1987)

9.51 J. E. Vinson, J. J. Liou: Electrostatic discharge in semiconductor devices: overview, Proc. IEEE **86**, 399–419 (1998)

9.52 N. Wan, K. Manning: *Exceeding 60-year life expectancy from an electronic energy meter*, Metering Asia Pac. Conf. (Kuala Lumpur, Malaysia 20–22 February 2001)

9.53 W. W. Abadeer, A. Bagramian, D. W. Conkle, C. W. Griffin, E. Langois, B. F. Lloyd, R. P. Mallette, J. E. Massucco, J. M. McKenna, S. W. Mittl, P. H. Noel: Key measurements of ultrathin gate dielectric reliability and in-line monitoring, IBM J. Res. Devel. **43**, 407–417 (1999)

9.54 C. Hu: Future CMOS scaling and reliability, Proc. IEEE **81**, 682–689 (May 1993)

9.55 J. Stevenson, J. A. Nachlas: *Microelectronics reliability prediction derived from component defect densities* (Pro Annual Reliab. Maintainab. Sympos., Los Angeles, CA 1990) pp. 366–370

9.56 R. Doyle, J. Barrett: Design for Reliability: a Focus on Electronics Assembly, Solder. Surf. Mount Technol. **9**(3), 22–29 (1997)

9.57 M. Baz, M. Udrea-Spenea, E. Tsoi, F. Turtudau, V. Ilian, G. Papaioannou, L. Galateanu, A. Stan, A. Tserepi, M. Bucur: Building in reliability technology for diodes manufacturing, IEEE CAS 2000 Proceedings **1**, 333–337 (2000)

9.58 W. T. K. Chien, C. H. J. Huang: Pracitical building in reliability approaches for semiconductor manufacturing, IEEE Trans. Reliab. **52**, 469–482 (2002)

9.59 S. Garrard: *Production implementation of a practical WLR program* (1994 IRW Final Report IEEE International, Lake Tahoe, CA 1995) pp. 20–29

9.60 L. N. Lie, A. K. Kapoor: *Wafer level reliability procedures to monitor gate oxide quality using V ramp and J ramp test methodology*, 1995 IRW Final Report (IEEE International, Gold Canyon, AZ 1996) pp. 113–121

9.61 A. P. Bieringer, D. Koch, H. Kammer, a. Kohlhase, A. Lill, A. Preusser, A. Schlemm, M. Schneegans: *Implementation of a WLR–program into a production line* (1995 IRW Final Report, Gold Canyon, AZ 1996) pp. 49–54

9.62 T. A. Dellin, W. M. Miller, D. G. Pierce, E. S. Snyder: Wafer level reliability, SPIE Microelectron. Manuf. Reliab. **18**(2), 144–154 (1992)

9.63 J. M. Soden, R. E. Anderson: IC failure analysis: techniques and tools for quality and reliability improvement, Proc. IEEE **81**, 703–715 (1993)

9.64 A. W. Righter, C. F. Hawkins, J. M. Soden, P. Maxwell: *CMOS IC reliability indicators and burn-in economics* (International Test Conference, IEEE Computer Society, Washington D.C. 1998) pp. 194–204

9.65 W. Kuo, Y. Kuo: Facing the headaches of early failures: a state-of-the-art review of burn-in decisions, Proc. IEEE **71**, 1257–1266 (1983)

9.66 A. Vassighi, O. Semenov, M. Sachdev: CMOS IC technology scaling and its impact on burn-in, IEEE Trans. Dev. Mater. Reliab. **4**, 208–222 (2004)

9.67 D. Gralian: Next generation burn-in development, IEEE Trans. Components Packaging Manuf. Technol. **17**, 190–196 (May 1994)

9.68 B. Vasquez, S. Lindsey: *The promise of known-good-die technologies, MCM '94 Proc.* (IEEE Society, Santa Clara, CA 1994) pp. 1–6

9.69 D. Inbar, M. Murin: KGD for flash memory: burn-in is key, Semicond. Int. **8**, 1 (2004)

9.70 A. Wager: *Wafer probe tests and stresses used for quality and reliability improvement*, 8th Annual KGD packaging and Test Workshop Proceedings (Die Products Consortium, Napa, CA September 2001)

9.71 P. Garrou: The wafer level packaging evolution, Semicond. Int. **27**, 119 (10/2/2004)

9.72 R. Nelson: Hot tests whip chips into shape, Test Meas. World **20**(12), 30 (October 2000)

9.73 D. Romanchik: Why burn-in ICs ?, Test Measurement World **12**, 85–88 (Oct. 1992)

9.74 J. Mi: Warranty policies and burn-in, Naval Res. Log. **44**, 199–209 (1997)

9.75 J. Mi: minimizing some cost functions related to both burn-in and field use, Oper. Res. **44**, 497–500 (1996)

9.76 S. H. Sheu, Y. H. Chien: |Minimizing cost functions related to both burn-in and field operation under a generalized model, IEEE Trans. Reliab. **53**, 435–440 (2004)

9.77 Kosznik, Drapella: Combining preventive replacements and burn-in procedures, Qual. Reliab. Eng. Int. **18**, 423–427 (2002)

9.78 J. H. Cha: On a better burn-in procedure, J. Appl. Probab. **37**, 1099–1103 (2000)

9.79 J. H. Cha: Burn-in procedure for a generalized model, J. Appl. Probab. **38**, 542–553 (2001)

9.80 J. H. Cha, S. Lee, J. Mi: Bonding the optimal burn-in time for a system with two types of failure, Naval Res. Logist. **51**, 1090–1101 (2004)

9.81 S. T. Tseng, C. Y. Peng: Optimal burn-in policy by using an integrated Wiener process, IIE Trans. **36**, 1161–1170 (2004)

9.82 K. N. Kim: Optimal burn-in for minimizing cost and multi-objectives, Microelectron. Reliab. **38**, 1577–1583 (1998)

9.83 W. T. K. Chien, W. Kuo: A nonparametric approach to estimate system burn-in time, IEEE Trans. Semicond. Manuf. **9**, 461–466 (1996)

9.84 W. T. K. Chien, W. Kuo: A nonparametric Bayes approach to decide system burn-in time, Naval Res. Logist. **44**, 655–671 (1997)

9.85 W. Kuo: Reliability enhancement through optimal burn-in, IEEE Trans. Reliab. **R-33**, 145–156 (1984)

9.86 D. H. Chi, W. Kuo: Burn-in optimization under reliability and capacity restrictions, IEEE Trans. Reliab. **38**, 193–199 (1989)

9.87 T. R. Kar, J. A. Nachlas: Coordinated warranty & burn-in strategies, IEEE Trans. Reliab. **46**, 512–519 (1997)

9.88 K. O. Kim, W. Kuo: Percentile residual life and system reliability as performance measures in the optimal system design, IIE Trans. **35**, 1133–1142 (2003)

9.89 H. W. Block, J. Mi, T. H. Savits: Some results on burn-in, Stat. Sin. **4**, 525–534 (1994)

9.90 H. W. Block, J. Mi, T. H. Savits: Burn-in at the component and system level. In: *Lifetime Data: Models in Reliability and Survival Analysis*, ed. by N. P. Jewell, A. C. Kimber, M. L. T. Lee, G. A. Whitmore (Kluwer, Dordrecht 1995)

9.91 C. W. Whitbeck, L. M. Leemis: Component vs. system burn-in techniques for electronic equipment, IEEE Trans. Reliab. **38**, 206–209 (1989)

9.92 R. K. Reddy, D. L. Dietrich: A 2-level environmental stress screening model: a mixed distribution approach, IEEE Trans. Reliab. **43**, 85–90 (1994)

9.93 E. A. Pohl, D. L. Dietrich: *Environmental stress screening strategies for multi-component systems with weibull failure times and imperfect failure detection* (Proceedings Annual Reliability and Maintainability Symposium, Washington D.C. 1995) pp. 223–232

9.94 W. Kuo: Incompatibility in evaluating large-scale systems reliability, IEEE Trans. Reliab. **43**, 659–660 (1994)

9.95 W. T. K. Chien, W. Kuo: Modeling and maximizing burn-in effectiveness, IEEE Trans. Reliab. **44**, 19–25 (1995)

9.96 W. T. K. Chien, W. Kuo: Optimal burn-in simulation on highly integrated circuit systems, IIE Trans. **24**, 33–43 (1992)

9.97 T. H. Kim, W. Kuo: Optimal burn-in decision making, Qual. Reliab. Eng. Int. **14**, 417–423 (1998)

9.98 K. O. Kim, W. Kuo: A general model of heterogeneous system lifetimes and conditions for system burn-in, Naval Res. Log. **50**, 364–380 (2003)

9.99 K. O. Kim, W. Kuo: Some considerations on system burn-in, IEEE Trans. Reliab. **54**(2), 207–214 (2005)

9.100 K. O. Kim, W. Kuo: Two-level burn-in for reliability and economy in repairable series systems having incompatibility, J. Reliab. Qual. Saf. Eng. **11**(3), 197–211 (2004)

9.101 F. Kuper, J. Vander Pol, E. Ooms, T. Johnson, R. Wijburg: *Relation between yield and reliability of integrated circuits: experimental results and application to continuous early failure rate reduction programs* (Proc. Int. Reliab. Phys. Symp., Dallas, Texas 1996) pp. 17–21

9.102 J. Van der Pol, F. Kuper, E. Ooms: Relation between yield and reliability of ICs and application to failure rate assessment and reduction in the one digit fit and ppm reliability era, Microelectron. Reliab. **36**, 1603–1610 (1996)

9.103 T. Zhao, Y. Hao, P. Ma, T. Chen: *Relation between reliability and yield of ICs based on discrete defect distribution model*, Proc. 2001 IEEE Int. Symp. Defect Fault Tolerance in VLSI systems (IEEE Society, San Francisco, CA 2001)

9.104 J. Van der Pol, E. Ooms, T. Hof, F. Kuper: *Impact of screening of latent defects at electrical test on yield-reliability relation and application to burn-in elimination* (Int. Relia. Phys. Sympos., Reno, NV 1998) pp. 370–377

9.105 K. R. Forbes, N. Arguello: Using time-dependent reliability fallout as a function of yield to optimize burn-in time for a 130nm SRAM device. In: *Integ. Reliab. Workshop Final Rep., IEEE Int., Lake Tahoe, CA* (IEEE Society 2003) pp. 61–66

9.106 T. H. Kim, W. Kuo, K. Chien: Burn-in effect on yield, IEEE Trans. Electron. Pack. Manuf. **23**, 293–299 (2000)

9.107 K. O. Kim, M. Zuo, W. Kuo: On the relationship of semiconductor yield and reliability, IEEE Trans. Semicond. Manuf. **18**(3), 422–429 (2005)

Part B Process Mo

Part B Process Monitoring and Improvement

Part B focuses on process monitoring, control and improvement. Chapter 10 describes in detail numerous important statistical methodologies for quality and productivity improvement, including statistical process control, robust design, signal-to-noise ratio, experimental design, and Taguchi methods. Chapter 11 deals with Six Sigma design and methodology. The chapter also discusses decision-making optimization strategies for product and process improvement, including design of experiments and the response-surface methodology. Chapter 12 describes the two widely used parameter-optimization techniques, the response-surface methodology and the Taguchi method, and discusses how to enhance existing methods by developing robust optimization approaches that better maximize the process and product performance. Chapter 13 introduces the concept of uniform design and its applications in the pharmaceutical industry and accelerated stress testing. It also discusses the methods of construction of uniform designs for experiments with mixtures in multidimensional cubes and some relationships between uniform designs and other related designs, while Chapt. 14 focuses on the development and applications of cumulative score statistics and describes the generalized theoretical development from traditional process-monitoring charts as well as how can they be applied to the monitoring of autocorrelated data. Chapter 15 provides a comprehensive review of various chain sampling plans such as acceptance sampling two-stage chains, dependent sampling, and chain sampling with variable inspection, and discusses several interesting extensions of chain sampling, including chain sampling for mixed attribute/variable inspection and deferred sampling plans. Chapter 16 surveys several major models and techniques, such as control charts based on the zero-inflated Poisson distribution, the generalized Poisson distribution and the time-between-event monitoring process, that can be used to monitor high quality processes. Chapter 17 introduces the basic concept and the use of the exponentially weighted moving-average statistic as a process-monitoring scheme commonly used for processes and maintenance in industrial plants. The chapter also discusses some recent innovative types of control charts. Chapter 18 provides a brief review of major univariate quality-monitoring procedures including Crosier's cumulative sum and exponentially weighted moving-average schemes and discusses various multivariate monitoring schemes for detecting a change in the level of a multivariate process.

10. Statistical Methods for Quality and Productivity Improvement

The first section of this chapter introduces statistical process control SPC and robust design RD, two important statistical methodologies for quality and productivity improvement. Section 10.1 describes in-depth SPC theory and tools for monitoring independent and autocorrelated data with a single quality characteristic. The relationship between SPC methods and automatic process control methods is discussed and differences in their philosophies, techniques, efficiencies, and design are contrasted. SPC methods for monitoring multivariate quality characteristics are also briefly reviewed.

Section 10.2 considers univariate RD, with emphasis on experimental design, performance measures and modeling of the latter. Combined and product arrays are featured and performance measures examined, include signal-to-noise ratios SNR, PerMIAs, process response, process variance and desirability functions. Of central importance is the decomposition of the expected value of squared-error loss into variance and off-target components which sometimes allows the dimensionality of the optimization problem to be reduced.

Section 10.3 deals with multivariate RD and demonstrates that the objective function for the multiple characteristic case is typically formed by additive or multiplicative combination of the univariate objective functions. Some alternative objective functions are examined as well as strategies for solving the optimization problem.

Section 10.4 defines dynamic RD and summarizes related publications in the statistics literature, including some very recent entries. Section 10.5 lists RD case studies originating from applications in manufacturing, reliability and tolerance design.

In the current international marketplace, continuous quality improvement is pivotal for maintaining a competitive advantage. Although quality improvement activities are most efficient and cost-effective when implemented as part of the design and development stage (off-line), on-line activities such as statistical process control (SPC) are vital for maintaining quality during manufacturing processes.

Statistical process control (SPC) is an effective tool for achieving process stability and improving process capability through variation reduction. Primarily, SPC is used to classify sources of process variation as either

common cause or assignable cause. Common cause variations are inherent to a process and can be described implicitly or explicitly by stochastic models. Assignable cause variations are unexpected and difficult to predict beforehand. The basic idea of SPC is to quickly detect and correct assignable cause variation before quality deteriorates and defective units are produced. The primary SPC tool was developed in the 1920s by Walter Shewhart of Bell Telephone Laboratories and has been tremendously successful in manufacturing applications [10.1–3].

Robust design is a systematic methodology that uses statistical experimental design to improve the design of products and processes. By making product and process performance insensitive (robust) to hard-to-control disturbances (noise), robust design simultaneously improves product quality, the manufacturing process, and reliability. The RD method was originally developed by the Japanese quality consultant, *Genichi Taguchi* [10.4]. Taguchi's 1980 introduction of robust parameter design to several major American industries resulted in significant quality improvements in product and process design [10.5]. Since then, a great deal of research on RD has improved related statistical techniques and clarified underlying principles.

In addition, many RD case studies have demonstrated phenomenal cost savings. In the electronics industry, *Kackar* and *Shoemaker* [10.6] reported a 60% process variance reduction; *Phadke* [10.5] reported a fourfold reduction in process variance and a twofold reduction in processing time – both from running simple RD experiments. In other industries, the American Supplier Institute (1983–1990) reported a large number of successful case studies in robust design.

Although most data is multivariate in nature, research in both areas has largely focused on normally distributed univariate characteristics (responses). *Montgomery* and *Woodall* [10.2] present a comprehensive panel discussion on SPC (see also *Woodall* and *Montgomery* [10.7]) and multivariate methods are reviewed by *Lowry* and *Montgomery* [10.8] and *Mason* [10.9]. Seminal research papers on RD include *Kackar* [10.10], *Leon* et al. [10.11], *Box* [10.12], *Nair* [10.13] and *Tsui* [10.14]. RD problems with multiple characteristics are investigated by *Logothetis* and *Haigh* [10.15], *Pignatiello* [10.16], *Elsayed* and *Chen* [10.17] and *Tsui* [10.18]. This research has yielded techniques allowing engineers to effectively implement SPC and RD in a host of applications.

This paper briefly revisits the major developments in both SPC and RD that have occurred over the last twenty years and suggests future research directions while highlighting multivariate approaches. Section 10.1 covers SPC of univariate and multivariate random variables for both Shewhart (including $\bar{x}$ and s charts) and non-Shewhart approaches (CUSUM and EWMA) while assessing the effects of autocorrelation and automatic process control. Section 10.2 considers univariate RD, emphasizing performance measures and modeling for loss functions, dual responses and desirability functions. Sections 10.3 and 10.4 deal respectively with multivariate and dynamic RD. Finally, Sect. 10.5 recaps RD case studies from the statistics literature in manufacturing, process control and tolerance design.

10.1 Statistical Process Control for Single Characteristics

The basic idea in statistical process control is a binary view of the state of a process; in other words, it is either running satisfactorily or not. *Shewhart* [10.19] asserted that the process state is related the type of variation manifesting itself in the process. There are two types of variation, called common cause and assignable or special cause variation. Common cause variation refers to the assumption that "future behavior can be predicted within probability limits determined by the common cause system" [10.20]. Special cause variation refers to "something special, not part of the system of common causes" [10.21]. A process that is subject only to common cause variation is "statistically" in control, since the variation is inherent to the process and therefore eliminated only with great difficulty. The objective of statistical process control is to identify and remove special cause variation as quickly as possible.

SPC charts essentially mimic a sequential hypothesis test to distinguish assignable cause variation from common cause variation. For example, a basic mathematical model behind SPC methods for detecting change in the mean is

$$X_t = \eta_t + Y_t \,,$$

where X_t is the measurement of the process variable at time t, and η_t is the process mean at that time. Here Y_t represents variation from the common cause system. In some applications, Y_t can be treated as an independently and identically distributed (iid) process. With few exceptions, the mean of the process is constant except

for abrupt changes, so

$$\eta_t = \eta + \mu_t \,,$$

where η is the mean target and μ_t is zero for $t < t_0$ and has nonzero values for $t \geq t_0$. For analytical simplicity step changes are often assumed; in other words μ_t remains at a new constant level μ for $t \geq t_0$.

10.1.1 SPC for i.i.d. Processes

The statistical goal of SPC control charts is to detect the change point t_0 as quickly as possible and trigger corrective action to bring the process back to the quality target. Among many others, the Shewhart chart, the EWMA chart, and the CUSUM chart are three important and widely used control charts.

Shewhart Chart

The Shewhart control chart monitors the process observations directly,

$$W_t = X_t - \eta \,.$$

Assuming that the standard deviation of W_t is σ_W, the stopping rule of the Shewhart chart is defined as $|W_t| > L\sigma_W$, where L is prespecified to maintain particular probability properties.

EWMA Chart

Roberts [10.22] introduces a control charting algorithm based on the exponentially weighted moving average of the observations,

$$W_t = \sum_{i=0}^{\infty} w_i (X_{t-i} - \eta) \,,$$

where $w_i = \lambda(1-\lambda)^i, (0 < \lambda \leq 1)$. It can be rewritten as

$$W_t = (1-\lambda) W_{t-1} + \lambda (X_t - \eta) \,, \qquad (10.1)$$

where $W_0 = 0$ or the process mean. The stopping rule of the EWMA chart is $|W_t| > L\sigma_W$ where $\sigma_W = \sqrt{\lambda/(2-\lambda)}\sigma_X$. The Shewhart chart is a special case of the EWMA chart with $\lambda = 1$. When the underlying process is i.i.d, the EWMA chart with small λ values is sensitive to the detection of small and medium shifts in mean [10.23].

CUSUM Chart

Page [10.24] introduces the CUSUM chart as a sequential probability test. It can be simply obtained by letting λ approach zero in (10.1). The CUSUM algorithm assigns equal weights to past observations, and its tabular form consists of two quantities,

$$W_t^+ = \max[0,\, W_{t-1}^+ + (X_t - \eta) - k\sigma_X] \,,$$
$$W_t^- = \min[0,\, W_{t-1}^- + (X_t - \eta) + k\sigma_X] \,,$$

where $W_0^+ = W_0^- = 0$. It can be shown that the CUSUM chart with $k = \mu/2$ is optimal for detecting a mean change in μ when the observations are i.i.d.

Because of the randomness of the observations, these control charts may trigger false alarms – out-of-control signals issued when the process is still in control. The expected number of units measured between two successive false alarms is called the *in-control* average run length $(\mathrm{ARL})_0$. When a special cause presents itself, the expected period before a signal is triggered is called the *out-of-control* average run length (ARL_1). The ideal control chart has a long ARL_0 and a short ARL_1. The Shewhart chart typically uses the constant $L = 3$ so that the in-control ARL is 370 when the underlying process is i.i.d. with normal distribution.

These SPC charts are very effective for monitoring the process mean when the process data is i.i.d. It has been shown that the Shewhart chart is sensitive for detecting large shifts while the EWMA and CUSUM charts are sensitive to small shifts [10.23]. However, a fundamental assumption behind these SPC charts is that the common cause variation is free of serial correlation. Due to the prevalence of advanced sensing and measurement technology in manufacturing processes, the assumption of independence is often invalid. For example, measuring critical in-process dimensions is now possible on every unit in the production of discrete parts. In continuous process production systems, the presence of inertial elements such as tanks, reactors, and recycle streams often result in significant serial correlation in the measured variables. Serial correlation creates many challenges and opportunities for SPC methodologies.

10.1.2 SPC for Autocorrelated Processes

Traditional SPC charts have been shown to function poorly while monitoring and controlling serially correlated processes [10.25, 26]. To accommodate autocorrelation, the following time series methods have been proposed.

Modifications of Traditional Methods

One common SPC strategy is to plot the autocorrelated data on traditional charts whose limits have been modified to account for the correlation. *Johnson* and

Bagshaw [10.27] and *Bagshaw* and *Johnson* [10.28] consider the effects of autocorrelation on CUSUM charts using the weak convergence of cumulative sums to a Wiener process. Another alternative is the exponentially weighted moving average chart for stationary processes (EWMAST) studied by *Zhang* [10.29]. *Jiang* et al. [10.30] extend this to a general class of control charts based on autoregressive moving average (ARMA) charts. The monitoring statistic of an ARMA chart is defined to be the result of a *generalized* ARMA(1, 1) process applied to the underlying process $\{X_t\}$,

$$\begin{aligned} W_t &= \theta_0 X_t - \theta X_{t-1} + \phi W_{t-1} \\ &= \theta_0 (X_t - \beta X_{t-1}) + \phi W_{t-1} \,, \end{aligned} \tag{10.2}$$

where $\beta = \theta/\theta_0$ and θ_0 is chosen so that the sum of the coefficients is unity when W_t is expressed in terms of the X_t's, so $\theta_0 = 1 + \theta - \phi$. The authors show that these charts exhibit good performance when the chart parameters are chosen appropriately.

Forecast-Based Monitoring Methods

Forecast-based charts started with the special-cause charts (SCC) proposed by *Alwan* and *Roberts* [10.31]. The general idea is to first apply a one-step-ahead predictor to the observation $\{X_t\}$ and then monitor the corresponding prediction error,

$$W_t = e_t \,, \tag{10.3}$$

where $e_t = X_t - \hat{X}_t$ is the forecast error of predictor $\hat{X}_t$. The SCC method is the first example that uses minimum mean squared error (MMSE) predictors and monitors the MMSE residuals. When the model is accurate, the MMSE prediction errors are approximately uncorrelated. This removal of correlation means that control limits for the SCC can be easily calculated from traditional Shewhart charts, EWMA charts, and CUSUM charts. Another advantage of the SCC method is that its performance can be analytically approximated.

The SCC method has attracted considerable attention and has been extended by many authors. Among them, *Harris* and *Ross* [10.25] and *Superville* and *Adams* [10.32] investigate process monitoring based on the MMSE prediction errors for simple autoregressive [AR(1)] models; *Wardell* et al. [10.33, 34] discuss the performance of SCC for ARMA(1, 1) models; and *Vander Wiel* [10.35] studies the performance of SCC for integrated moving average [IMA(0, 1, 1)] models. SCC methods perform poorly when detecting small shifts since a constant mean shift always results in a dynamic shift pattern in the error term.

In general this approach can be applied to any predictor. *Montgomery* and *Mastrangelo* [10.36] recommend the use of EWMA predictors in the SCC method (hereafter called the M–M chart). *Jiang* et al. [10.37] propose the use of proportional-integral-derivative (PID) predictors

$$\begin{aligned} \hat{X}_t = \hat{X}_{t-1} &+ (k_\mathrm{P} + k_\mathrm{I} + k_\mathrm{D}) e_{t-1} \\ &- (k_\mathrm{P} + 2k_\mathrm{D}) e_{t-2} + k_\mathrm{D} e_{t-3} \,, \end{aligned} \tag{10.4}$$

where k_P, k_I, and k_D are parameters of the PID controller defined in Sect. 10.1.3. The family of PID-based charts includes the SCC, EWMA, and M–M charts as special cases. *Jiang* et al. [10.37] show that the predictors of the EWMA chart and M–M chart may sometimes be inefficient and the SCC over-sensitive to model deviation. They also show that the performance of the PID-based chart is affected by the choice of chart parameters. For any given underlying process, one can therefore tune the parameters of the PID-based chart to optimize its performance.

GLRT-Based Multivariate Methods

Since forecast-based residual methods monitor a single statistic e_t, they often suffer from the problem of a narrow "window of opportunity" when the underlying process is positively correlated [10.35]. If the shift occurrence time is known, the problem can be alleviated by including more historical observations/residuals in the test. This idea was first proposed by *Vander Wiel* [10.35] using a generalized likelihood ratio test (GLRT) procedure. Assuming residual signatures $\{\delta_i\}$ when a shift occurs, the GLRT procedure based on residuals is

$$W_t = \max_{0 \le k \le p-1} \left| \sum_{i=0}^{k} \delta_i e_{t-k+i} \right| \Big/ \sqrt{\sum_{i=0}^{k} \delta_i^2} \,, \tag{10.5}$$

where p is the prespecified size of the test window. *Apley* and *Shi* [10.38] show that this procedure is very efficient in detecting mean shifts when p is sufficiently large. Similar to the SCC methods, this is model-based and the accuracy of signature strongly depends on the window length p. If p is too small and a shift is not detected within the test window, the signature in (10.5) might no longer be valid and the test statistic no longer efficient.

Note that a step mean shift at time $t-k+1$ results in a signature

$$d_k = (0, \cdots, 0, \overbrace{1, \cdots, 1}^{k})' \quad (1 \le k \le p)$$

and

$$d_k = (1, 1, \cdots, 1)' \quad (k > p)$$

on $U_t = (X_{t-p+1}, X_{t-p+2}, \cdots, X_t)'$. To test these signatures, the GLRT procedure based on observation vector W_t is defined as

$$W_t = \max_{0 \le k \le p-1} |d_k' \Sigma_U^{-1} U_t| / \sqrt{d_k' \Sigma_U^{-1} d_k} \,, \qquad (10.6)$$

where Σ_U is the covariance matrix of U_t. *Jiang* [10.39] points out that this GLRT procedure is essentially model-free and always matches the true signature of U_t regardless of the timing of the change point. If a non-step shift in the mean occurs, multivariate charts such as Hotelling's T^2 charts can be developed accordingly [10.40].

Monitoring Batch Means

One of the difficulties with monitoring autocorrelated data is accounting for the underlying autocorrelation. In simulation studies, it is well known that batch means reduce autocorrelation within data. Motivated by this idea, *Runger* and *Willemain* [10.41, 42] use a weighted batch mean (WBM) and a unified batch mean (UBM) to monitor autocorrelated data. The WBM method weighs the mean of observations, defines batch size so that autocorrelation among batches is reduced to zero and requires knowledge of the underlying process model [10.43]. The UBM method determines batch size so that autocorrelation remains below a certain level and is "model-free". Runger and Willemain show that the UBM method is simple and often more cost-effective in practice.

Batch-means methods not only develop statistics based on batch-means, but also provide variance estimation of these statistics for some commonly used SPC charts. *Alexopoulos* et al. [10.44] discuss promising methods for dealing with correlated observations including nonoverlapping batch means (NBM), overlapping batch means (OBM) and standardized time series (STS).

10.1.3 SPC versus APC

Automatic process control (APC) complements SPC as a variation reduction tool for manufacturing industries. While SPC techniques are used to reduce unexpected process variation by detecting and removing the cause of variation, APC techniques are used to reduce systematic variation by employing feedforward and feedback control schemes. The relationships between SPC and APC are important to both control engineers and quality engineers.

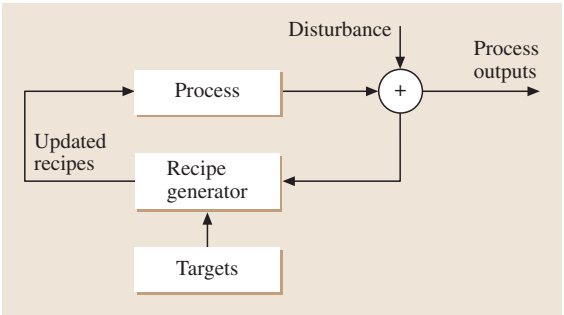


Fig. 10.1 Automatic process control

Feedback Control versus Prediction

The feedback control scheme is a popular APC strategy that uses the deviation of output from target (set-point) to signal a disturbance of the process. This deviation or error is then used to compensate for the disturbance. Consider a pure-gain dynamic feedback-controlled process, as shown in Fig. 10.1. The process output can be expressed as

$$e_t = X_t - Z_{t-1} \,. \qquad (10.7)$$

Suppose $\hat{X}_t$ is an estimator (a predictor) of X_t that can be obtained at time $t-1$. A realizable form of control can be obtained by setting

$$Z_{t-1} = -\hat{X}_t \qquad (10.8)$$

so that the output error at time $t+1$ becomes

$$e_t = X_t - \hat{X}_t \,, \qquad (10.9)$$

which is equal to the "prediction error". Control and prediction can therefore have a one-to-one corresponding relationship via (10.8) and (10.9).

As shown in *Box* and *Jenkins* [10.45], when the process can be described by an ARIMA model, the MMSE control and the MMSE predictor have exactly the same form. Serving as an alternative to the MMSE predictor, the EWMA predictor corresponds to the integral (I) control [10.46] and is one of the most frequently used prediction methods due to its simplicity and efficiency. In general, the EWMA predictor is robust against nonstationarity due to the fact that the I control can continuously adjust the process whenever there is an offset.

An extension of the I control is the widely used PID control scheme,

$$Z_t = -k_P e_t - k_I \frac{1}{1-B} e_t - k_D (1-B) e_t \,, \qquad (10.10)$$

where k_P, k_I, and k_D are constants that, respectively, determine the amount of proportional, integral, and derivative control action. The corresponding PID predictor (10.4) can be obtained from (10.8) and (10.10). When $\lambda_3 = 0$, in other words when $k_D = 0$ (and thus $\lambda_1 = k_P + k_I$ and $\lambda_2 = -k_P$), we have a PI predictor corresponding to the proportional-integral control scheme commonly used in industry.

Process Prediction versus Forecast-Based Monitoring Methods

As discussed in Sect. 10.1.2, one class of SPC methods for autocorrelated processes starts from the idea of "whitening" the process and then monitoring the "whitened" process with time series prediction models. The SCC method monitors MMSE prediction errors and the M–M chart monitors the EWMA prediction error. Although the EWMA predictor is optimal for an IMA(0, 1, 1) process, the prediction error is no longer i.i.d. for predicting other processes. Most importantly, the EWMA prediction error that originated from the I control can compensate for mean shifts in steady state which makes the M–M chart very problematic for detecting small shifts in mean.

Since PID control is very efficient and robust, PID-based charts motivated by PID predictors outperform SCC and M–M charts. APC-based knowledge of the process can moreover clarify the performance of PID-based charts. In summary, the P term ensures that process output is close to the set point and thus sensitive in SPC monitoring, whereas the I term always yields control action regardless of error size which leads to a zero level of steady-state error. This implies that the I term is dominant in SPC monitoring. The purpose of derivative action in PID control is to improve closed-loop stability by making the D term in SPC monitoring less sensitive. Although there is no connection between the EWMA predictor and the EWMA chart, it is important to note that the I control leads to the EWMA predictor and the EWMA prediction-based chart is the M–M chart. As shown in *Jiang* et al. [10.37], the EWMA chart is the same as the P-based chart.

10.1.4 SPC for Automatically Controlled Processes

Although APC and SPC techniques share the objective of reducing process variation, their advocates have quarrelled for decades. It has recently been recognized that the two techniques can be integrated to produce more efficient tools for process variation reduction [10.47–52].

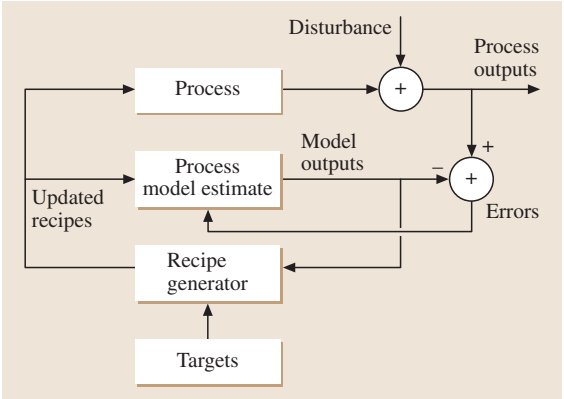


Fig. 10.2 APC/SPC integration

This APC/SPC integration employs an APC rule to regulate the system and superimposes SPC charts on the APC-controlled system to detect process departures from the system model. Using Deming's terminology, the APC scheme is responsible for reducing common cause variation while the SPC charts are responsible for reducing assignable cause variation. From the statistical point of view, the former part resembles a parameter estimation problem for forecasting and adjusting the process and the latter part emulates a hypothesis test of process location. Figure 10.2 pictures a conceptual integration of SPC charts into the framework of a feedback control scheme. To avoid confusion, *Box* and *Luceno* [10.46] refer to APC activities as process adjustment and to SPC activities as process monitoring. Since this chapter emphasizes SPC methods for quality improvement, we discuss only the monitoring component of APC/SPC integration.

As discussed in Sect. 10.1.3, control charts developed for monitoring autocorrelated observations shed light on the monitoring of integrated APC/SPC systems. Fundamentally, the output of an automatically controlled process is recommended for SPC monitoring. This is equivalent to forecast-based control charts of the corresponding predictor. For example, if the process is controlled by an MMSE controller, monitoring the output is exactly the same as the SCC method. Similar to forecast-based methods, assignable causes have an effect that is always contaminated by the APC control action which results in a limited "window of opportunity" for detection [10.35]. As an alternative, some authors suggest that monitoring the APC control action may improve the probability of detection [10.20]. *Jiang* and *Tsui* [10.53] compare the performance of monitoring the output vs. the control action of an APC process and

show that for some autocorrelated processes monitoring the control action may be more efficient than monitoring the output of the APC system.

In general, the performance achieved by SPC monitoring an APC process depends on the data stream (the output or the control action) being measured, the APC control scheme employed, and the underlying autocorrelation of the process. If information from process output and control action can be combined, a universal monitor with higher SPC efficiency [10.51] can be developed. *Kourti* et al. [10.54] propose a method of monitoring process outputs conditional on the inputs or other changing process parameters. *Tsung* et al. [10.55] propose multivariate control charts such as Hotelling's T^2 chart and the Bonferroni approach to monitor output and control action simultaneously. Defining the vector of outputs and control actions as $V_t = (e_t, \cdots, e_{t-p+1}, X_t, \cdots, X_{t-p+1})'$, a dynamic T^2 chart with window size p monitors statistic

$$W_t = V_t' \Sigma_V^{-1} V_t \,,$$

where Σ_V is the covariance matrix of V_t [10.56]. W_t follows a χ^2 distribution during each period given known process parameters. However, strong serial correlation exists so that the χ^2 quantiles cannot be used for control limits. By recognizing the mean shift patterns of V_t, *Jiang* [10.57] develops a GLRT procedure based on V_t. This GLRT procedure is basically univariate and more efficient than the T^2 chart.

10.1.5 Design of SPC Methods: Efficiency versus Robustness

Among many others, the minimization of mean squared error/prediction error is one of the important criteria for prediction/control scheme design. Although the special cause chart is motivated by MMSE prediction/control, many previously mentioned SPC charts such as the PID chart have fundamentally different criteria from those of the corresponding APC controllers. When selecting SPC charts, the desired goal is maximization of the probability of shift detection.

For autocorrelated processes, *Jiang* [10.37] propose an ad hoc design procedure using PID charts. They demonstrate how two capability indices defined by signal-to-noise ratios (SNR) play a critical role in the evaluation of SPC charts. They denote σ_W as the standard deviation of charting statistic W_t and μ_T ($/\mu_S$) as the shift levels of W_t at the first step (/long enough) after the shift takes place. The transient state ratio is defined as $C_T = \mu_T/\sigma_W$, which measures the capability of the control chart to detect a shift in its first few steps. The steady state ratio is defined as $C_S = \mu_S/\sigma_W$, which measures the ability of the control chart to detect a shift in its steady state. These two signal-to-noise ratios determine the efficiency of the SPC chart and can be manipulated by selecting control chart parameters.

For a particular mean shift level, if the transient state ratio/capability can be tuned to a high value (say 4 to 5) by choosing appropriate chart parameters, the corresponding chart will detect the shift very quickly. Otherwise the shift will likely be missed during the transient state and will need to be detected in later runs. Since a high steady state ratio/capability heralds efficient shift detection at steady state, a high steady state ratio/capability is also desired. However, the steady state ratio/capability should not be tuned so high that it results in an extremely small transient ratio/capability, indicative of low probability of detection during the transient state. To endow the chart with efficient detection at both states, a tradeoff is needed when choosing the charting parameters. An approximate C_S value of 3 is generally appropriate for balancing the values of C_T and C_S.

One of the considerations when choosing an SPC method is its robustness to autocorrelated and automatically controlled processes. Robustness of a control chart refers to how insensitive its statistical properties are to model mis-specification. Reliable estimates of process variation are of vital importance for the proper functioning of all SPC methods [10.58]. For process X_t with positive first-lag autocorrelation, the standard deviation derived from moving range is often underestimated because

$$E(\hat{\sigma}_{MR}) = E(\overline{MR}/d_2) = \sigma_X \sqrt{1-\rho_1} \,,$$

where ρ_1 is the first-lag correlation coefficient of X_t [10.59].

A more serious problem with higher sensitivity control charts such as the PID chart is that they may be less robust than lower sensitivity control charts such as the SCC. *Tsung* et al. [10.60] and *Luceno* [10.61] conclude that PID controllers are generally more robust than MMSE controllers against model specification error. However *Jiang* [10.37] shows that PID charts tend to have a shorter "in-control" ARL when the process model is mis-specified since model errors can be viewed as a kind of "shift" from the "true" process model. This seems to be a discouraging result for higher sensitivity control charts. In practice, a trade-off is necessary between sensitivity and robustness when selecting control charts for autocorrelated processes. *Apley* and *Lee* [10.62] recommend using

a conservative control limit for EWMA charts when monitoring MMSE residuals. By using the worst-case estimation of residual variance, the EWMA chart can be robustly designed for the in-control state with a slight efficiency loss in the out-of-control state. This design strategy can be easily generalized to other SPC methods for autocorrelated or automatically controlled processes.

10.1.6 SPC for Multivariate Characteristics

Through modern sensing technology that allows frequent measurement of key quality characteristics during manufacturing, many in-process measurements are strongly correlated to each other. This is especially true for measurements related to safety, fault detection and diagnosis, quality control and process control. In an automatically controlled process for example, process outputs are often strongly related to process control actions. Joint monitoring of these correlated characteristics ensures appropriate control of the overall process. Multivariate SPC techniques have recently been applied to novel fields such as environmental monitoring and detection of computer intrusion.

The purpose of multivariate on-line techniques is to investigate whether measured characteristics are simultaneously in statistical control. A specific multivariate quality control problem is to consider whether an observed vector of measurements $\boldsymbol{x} = (x_1, \ldots, x_k)$ exhibits a shift from a set of "standard" parameters $\boldsymbol{\mu}^0 = (\mu_1^0, \ldots, \mu_k^0)'$. The individual measurements will frequently be correlated, meaning that their covariance matrix Σ will not be diagonal.

Versions of the univariate Shewhart, EWMA and CUSUM charts have been developed for the case of multivariate normality.

Multivariate T^2 Chart

To monitor a multivariate vector, *Hotelling* [10.63] suggested an aggregated statistic equivalent to the Shewhart control chart in the univariate case,

$$T^2 = (\boldsymbol{x} - \boldsymbol{\mu}^0)' \, \hat{\Sigma}_{\boldsymbol{x}}^{-1} \, (\boldsymbol{x} - \boldsymbol{\mu}^0) \,, \tag{10.11}$$

where $\hat{\Sigma}_{\boldsymbol{x}}$ is an estimate of the population covariance matrix Σ. If the population covariance matrix is known, Hotelling's T^2 statistic follows a χ^2 distribution with k degrees of freedom when the process is in-control. A signal is triggered when $\chi^2 > \chi^2_{k,\alpha}$. One of the important features of the T^2 charts is that its out-of-control performance depends solely on the noncentrality parameter $\delta = \sqrt{(\boldsymbol{\mu} - \boldsymbol{\mu}^0)' \Sigma_{\boldsymbol{x}}^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}^0)}$, where $\boldsymbol{\mu}$ is the actual mean vector. This means that its detectional performance is invariant along the contours of the multivariate normal distribution.

Multivariate EWMA Chart

Hotelling's T^2 chart essentially utilizes only current process information. To incorporate recent historical information, *Lowry* [10.64] develop a similar multivariate EWMA chart

$$W_t^2 = \boldsymbol{w}_t' \Sigma_{\boldsymbol{w}}^{-1} \boldsymbol{w}_t \,,$$

where $\boldsymbol{w}_t = \Lambda(\boldsymbol{x}_t - \boldsymbol{\mu}^0) + (\boldsymbol{I} - \Lambda)\boldsymbol{w}_{t-1}$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_k)$. For simplicity, $\lambda_i = \lambda \; (1 \le i \le k)$ is generally adopted and $\Sigma_{\boldsymbol{w}} = \lambda/(2-\lambda)\Sigma_{\boldsymbol{x}}$.

Multivariate CUSUM Chart

There are many CUSUM procedures for multivariate data. *Crosier* [10.65] proposes two multivariate CUSUM procedures, cumulative sum of T (COT) and MCUSUM. The MCUSUM chart is based on the statistics

$$\boldsymbol{s}_t = \begin{cases} \boldsymbol{0} & \text{if } C_t \le k_1 \\ (\boldsymbol{s}_{t-1} + \boldsymbol{x}_t)(1 - k_1/C_t) & \text{if } C_t > k_1 \,, \end{cases} \tag{10.12}$$

where $\boldsymbol{s}_0 = \boldsymbol{0}$, $C_t = \sqrt{(\boldsymbol{s}_{t-1} + \boldsymbol{x}_t)' \Sigma_{\boldsymbol{x}}^{-1} (\boldsymbol{s}_{t-1} + \boldsymbol{x}_t)}$, and $k_1 > 0$. The MCUSUM chart signals when $W_t = \boldsymbol{s}_t' \Sigma_{\boldsymbol{x}}^{-1} \boldsymbol{s}_t > h_1$. *Pignatiello* and *Runger* [10.66] propose another multivariate CUSUM chart (MC1) based on the vector of cumulative sums,

$$W_t = \max\left(0, \sqrt{D_t' \Sigma_{\boldsymbol{x}}^{-1} D_t} - k_2 l_t\right) \,, \tag{10.13}$$

where $k_2 > 0$, $D_t = \sum_{i=t-l_t+1}^{t} \boldsymbol{x}_i$, and

$$l_t = \begin{cases} l_{t-1} + 1 & \text{if } W_{t-1} > 0 \\ 1 & \text{otherwise} \,. \end{cases}$$

Once an out-of-control signal is triggered from a multivariate control chart, it is important to track the cause of the signal so that the process can be improved. Fault diagnosis can be implemented by T^2 decompositions following the signal and large components are suspected to be faulty. Orthogonal decompositions such as principal component analysis [10.67] are popular tools. *Mason* et al. [10.68], *Hawkins* [10.69] and *Hayter* and *Tsui* [10.70] propose other alternatives which integrate process monitoring and fault diagnosis. *Jiang* and *Tsui* [10.71] provide a thorough review of these methods.

10.2 Robust Design for Single Responses

Taguchi [10.4] introduced parameter design, a method for designing processes that are robust (insensitive) to uncontrollable variation, to a number of American corporations. The objective of this methodology is to find the settings of design variables that minimize the expected value of squared-error loss defined as

$$L(Y,t) = (Y-t)^2 \,, \tag{10.14}$$

where Y represents the actual process response and t the targeted value. A loss occurs if the response Y deviates from its target t. This loss function originally became popular in estimation problems considering unbiased estimators of unknown parameters. The expected value of $(Y-t)^2$ can be easily expressed as

$$\begin{aligned} E(L) &= A_0 E(Y-t)^2 \\ &= A_0\left[\mathrm{Var}(Y) + (E(Y)-t)^2\right] , \end{aligned} \tag{10.15}$$

where $\mathrm{Var}(Y)$ and $E(Y)$ are the mean and variance of the process response and A_0 is a proportional constant representing the economic costs of the squared error loss. If $E(Y)$ is on target then the squared-error loss function reduces to the process variance. Its similarity to the criterion of least squares in estimation problems makes the squared-error loss function easy for statisticians and engineers to grasp. Furthermore the calculations for most decision analyses based on squared-error loss are straightforward and easily seen as a trade-off between variance and the square of the off-target factor.

Robust design (RD) assumes that the appropriate performance measure can be modeled as a transfer function of the fixed control variables and the random noise variables of the process as follows:

$$Y = f(\boldsymbol{x}, \boldsymbol{N}, \boldsymbol{\theta}) + \epsilon \,, \tag{10.16}$$

where $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ is the vector of control factors, $\boldsymbol{N} = (N_1, \ldots, N_q)^T$ is the vector of noise factors, $\boldsymbol{\theta}$ is the vector of unknown response model parameters, and f is the transfer function for Y. The control factors are assumed to be fixed and represent the fixed design variables. The noise factors $\boldsymbol{N}$ are assumed to be random and represent the uncontrolled sources of variability in production. The pure error ϵ represents the remaining variability that is not captured by the noise factors, and is assumed to be normally distributed with zero mean and finite variance.

Taguchi divides the design variables into two subsets, $\boldsymbol{x} = (\boldsymbol{x}_\mathrm{a}, \boldsymbol{x}_\mathrm{d})$, where $\boldsymbol{x}_\mathrm{a}$ and $\boldsymbol{x}_\mathrm{d}$ are called respectively the adjustment and nonadjustment design factors. An adjustment factor influences process location while remaining effectively independent of process variation. A nonadjustment factor influences process variation.

10.2.1 Experimental Designs for Parameter Design

Taguchi's Product Arrays and Combined Arrays

Taguchi's experimental design takes an orthogonal array for the controllable design parameters (an inner array of control factors) and crosses it with another orthogonal array for the factors beyond reasonable control (an outer array of noise factors). At each test combination of control factor levels, the entire noise array is run and a performance measure is calculated. Hereafter we refer to this design as the product array. These designs have been criticized by *Box* [10.12] and others for being unnecessarily large.

Welch [10.72] combined columns representing the control and noise variables within the same orthogonal array. These combined arrays typically have a shorter number of test runs and do not replicate the design. The lack of replication prevents unbiased estimation of random error but we will later discuss research addressing this limitation.

Which to Use: Product Array or Combined Array. There is a wide variety of expert opinion regarding choice of experimental design in *Nair* [10.13]. The following references complement Nair's comprehensive discussion. *Ghosh* and *Derderian* [10.73] derive robustness measures for both product and combined arrays, allowing the experimenter to objectively decide which array provides a more robust option. *Miller* et al. [10.74] consider the use of a product array on gear pinion data. *Lucas* [10.75] concludes that the use of classical, statistically designed experiments can achieve the same or better results than Taguchi's product arrays. *Rosenbaum* [10.76] reinforces the efficiency claims of the combined array by giving a number of combined array designs which are smaller for a given orthogonal array strength or stronger for a given size. Finally, *Wu* and *Hamada* [10.77] provide an intuitive approach to choosing between product and combined array based on an effect-ordering principle.

They list the most important class of effects as those containing control–noise interactions, control main effects and noise main effects. The second highest class contains the control–control interactions and the control–control–noise interactions while the third and

least important class contains the noise–noise interactions. That array producing the highest number of clear effect estimates in the most important class is considered the best design.

Noting that the combined array is often touted as being more cost-effective due to an implied smaller number of runs, Wu and Hamada place the cost comparison on a more objective basis by factoring in both cost per control setting and cost per noise replicate. They conclude that the experimenter must prioritize the effects to be estimated and the realistic costs involved before deciding which type of array is optimal.

Choosing the Right Orthogonal Array for RD

Whether the experimenter chooses a combined or product array, selecting the best orthogonal array is an important consideration. The traditional practice in classical design of experiments is to pick a Resolution IV or higher design so that individual factors are aliased with three factor interactions, of which there are relatively few known physical examples.

However, the estimation of main effects is not necessarily the best way to judge the value of a test design for RD. The control–noise interactions are generally regarded as having equal importance as the control effects for fine tuning the final control factor settings for minimal product variation. Hence evaluation of an experimental design for RD purposes must take into account the design's ability to estimate the control–noise interactions deemed most likely to affect product performance.

Kackar and *Tsui* [10.78] feature a graphical technique for showing the confounding pattern of effects within a two-level fractional factorial. *Kackar* et al. [10.79] define orthogonal arrays and describe how Taguchi's fixed element arrays are related to well known fractional factorial designs. Other pieces related to this decision are *Hou* and *Wu* [10.80], *Berube* and *Nair* [10.60] and *Bingham* and *Sitter* [10.81].

D-Optimal Designs

In this section several authors show how D-optimal designs can be exploited in RD experiments. A D-optimal design minimizes the area of the confidence ellipsoids for parameters being estimated from an assumed model. Their key strength is their invariance to linear transformation of model terms and their characteristic weakness is a dependence on the accuracy of the assumed model. By using a proper prior distribution to attack the singular design problem and make the design less model-dependent, *Dumouchel* and *Jones* [10.82] provide a Bayesian D-optimal design needing little modification of existing D-optimal search algorithms.

Atkinson and *Cook* [10.83] extend the existing theory of D-optimal design to linear models with nonconstant variance. With a Bayesian approach they create a compromise design that approximates preposterior loss. *Vining* and *Schaub* [10.84] use D-optimality to evaluate separate linear models for process mean and variance. Their comparison of the designs indicates that replicated fractional factorials of assumed constant variance best estimate variance while semi-Bayesian designs better estimate process response.

Chang [10.85] proposes an algorithm for generating near D-optimal designs for multiple response surface models. This algorithm differs from existing approaches in that it does not require prior knowledge or data based estimates of the covariance matrix to generate its designs. *Mays* [10.86] extends the quadratic model methodology of RSM to the case of heterogeneous variance by using the optimality criteria D (maximal determinant) and I (minimal integrated prediction variance) to allocate test runs to locations within a central composite design.

Other Designs

The remaining references discuss types of designs used in RD which are not easily classified under the more common categories previously discussed.

Pledger [10.87] divides noise variables into observable and unobservable and argues that one's ability to observe selected noise variables in production should translate into better choices of optimal control settings. *Rosenbaum* [10.88] uses blocking to separate the control and noise variables in combined arrays, which were shown in *Rosenbaum* [10.76] to be stronger for a given size than the corresponding product array designs. *Li* and *Nachtsheim* [10.89] present experimental designs which don't depend on the experimenter's prior determination of which interactions are most likely significant.

10.2.2 Performance Measures in RD

In Sect. 10.2.1 we compared some of the experimental designs used in parameter design. Of equal importance is choosing which performance measure will best achieve the desired optimization goal.

Taguchi's Signal-to-Noise Ratios

Taguchi introduced a family of performance measures called signal-to-noise ratios whose specific form depends on the desired response outcome. The case where

the response has a fixed nonzero target is called the nominal-the-best case (NTB). Likewise, the cases where the response has a smaller-the-better target or a larger-the-better target are, respectively, called the STB and LTB cases.

To accomplish the objective of minimal expected squared-error loss for the NTB case, Taguchi proposed the following two-step optimization procedure: (i) Calculate and model the SNRs and find the nonadjustment factor settings which maximize the SNR. (ii) Shift mean response to the target by changing the adjustment factor(s).

For the STB and LTB cases, Taguchi recommends directly searching for the values of the design vector $\boldsymbol{x}$ which maximize the respective SNR. Alternatives for these cases are provided by *Tsui* and *Li* [10.90] and *Berube* and *Wu* [10.91].

Performance Measure Independent of Adjustment (PerMIAs)

Taguchi did not demonstrate how minimizing the SNR would achieve the stated goal of minimal average squared-error loss. *Leon* et al. [10.11] defined a function called the performance measure independent of adjustment (PerMIA) which justified the use of a two-step optimization procedure. They also showed that Taguchi's SNR for the NTB case is a PerMIA when both an adjustment factor exists and the process response transfer function is of a specific multiplicative form. When Taguchi's SNR complies with the properties of a PerMIA, his two-step procedure minimizes the squared-error loss.

Leon et al. [10.11] also emphasized two major advantages of the two-step procedure:

- It reduces the dimension of the original optimization problem.
- It does not require reoptimization for future changes of the target value.

Box [10.12] agrees with *Leon* et al. [10.11] that the SNR is only appropriately used in concert with models where process sigma is proportional to process mean. *Maghsoodloo* [10.92] derives and tabulates exact mathematical relationships between Taguchi's STB and LTB measures and his quality loss function.

Leon and *Wu* [10.93] extend the PerMIA of *Leon* et al. [10.11] to a maximal PerMIA which can solve constrained minimization problems in a two-step procedure similar to that of Taguchi. For nonquadratic loss functions, they introduce general dispersion, location and off-target measures while developing a two-step process. They apply these new techniques in a number of examples featuring additive and multiplicative models with nonquadratic loss functions. *Tsui* and *Li* [10.90] establish a multistep procedure for the STB and LTB problem based on the response model approach under certain conditions.

Process Response and Variance as Performance Measures

The dual response approach is a way of finding the optimal design settings for a univariate response without the need to use a loss function. Its name comes from its treatment of mean and variance as responses of interest which are individually modeled. It optimizes a primary response while holding the secondary response at some acceptable value.

Nair and *Pregibon* [10.94] suggest using outlier-robust measures of location and dispersion such as median (location) and interquartile range (dispersion). *Vining* and *Myers* [10.95] applied the dual response approach to Taguchi's three SNRs while restricting the search area to a spherical region of limited radius. *Copeland* and *Nelson* [10.96] solve the dual response optimization problem with the technique of direct function minimization. They use the Nelder-Mead simplex procedure and apply it to the LTB, STB and NTB cases. Other noteworthy papers on the dual response method include *Del Castillo* and *Montgomery* [10.97] and *Lin* and *Tu* [10.98].

Desirability as a Performance Measure

The direct conceptual opposite of a loss function, a utility function maps a specific set of design variable settings to an expected utility value (value or worth of a process response). Once the utility function is established, nonlinear direct search methods are used to find the vector of design variable settings that maximizes utility.

Harrington [10.99] introduced a univariate utility function called the desirability function, which gives a quality value between zero (unacceptable quality) and one (further improvement would be of no value) of a quality characteristic of a product or process. He defined the two-sided desirability function as follows:

$$d_i = \mathrm{e}^{-|Y_i'|^c} , \tag{10.17}$$

where e is the natural logarithm constant, c is a positive number subjectively chosen for curve scaling, and Y_i' is a linear transformation of the univariate response Y_i whose properties link the desirability values to product specifications. It is of special interest to note that for $c = 2$, a mid-specification target and response values

within the specification limits, this desirability function is simply the natural logarithm constant raised to the squared-error loss function.

Other Performance Measures

Ng and *Tsui* [10.100] derive a measure called q-yield which accounts for variation from target among passed units as well as nonconforming units. It does this by penalizing yield commensurate with the amount of variation measured within the passed units. *Moorhead* and *Wu* [10.91] develop modeling and analysis strategies for a general loss function where the quality characteristic follows a location-scale model. Their three-step procedure includes an adjustment step which moves the mean to the side of the target with lower cost. Additional performance measures are introduced in *Joseph* and *Wu* [10.101] and *Joseph* and *Wu* [10.102].

10.2.3 Modeling the Performance Measure

The third important decision the experimenter must grapple with is how to model the chosen performance measure. Linear models are by far the most common way to approximate loss functions, SNR's and product responses. This section covers response surface models, the generalized linear model and Bayesian modeling.

Response Surface Models

Response surface models (RSM) are typically second-order linear models with interactions between the first-order model terms. While many phenomena cannot be accurately represented by a quadratic model, the second-order approximation of the response in specific regions of optimal performance may be very insightful to the product designer.

Myers et al. [10.103] make the case for implementing Taguchi's philosophy within a well established, sequential body of empirical experimentation, RSM. The combined array is compared to the product array and the modeling of SNR compared to separate models for mean and variance. In addition, RSM lends itself to the use of mixed models for random noise variables and fixed control variables. *Myers* et al. [10.104] incorporate noise variables and show how mean and variance response surfaces can be combined to create prediction limits on future response.

Analysis of Unreplicated Experiments. The most commonly cited advantage of modeling process responses rather than SNR is the use of more efficient combined arrays. However the gain in efficiency usually assumes there is no replication for estimating random error. Here we review references for analyzing the data from unreplicated fractional factorial designs.

Box and *Meyer* [10.105] present an analysis technique which complements normal probability plots for identifying significant effects from an unreplicated design. Their Bayesian approach assesses the size of contrasts by computing a posterior probability that each contrast is active. They start with a prior probability of activity and assume normality of the significant effects and deliver a nonzero posterior probability for each effect.

Lenth [10.106] introduces a computationally simple and intuitively pleasing technique for measuring the size of contrasts in unreplicated fractional factorials. The Lenth method uses standard T statistics and contrast plots to indicate the size and significance of the contrast. Because of its elegant simplicity, the method of Lenth is commonly cited in RD case studies.

Pan [10.107] shows how failure to identify even small and moderate location effects can subsequently impair the correct identification of dispersion effects when analyzing data from unreplicated fractional factorials. *Ye* and *Hamada* [10.77] propose a simple simulation method for estimating the critical values employed by Lenth in his method for testing significance of effects in unreplicated fractional factorial designs.

McGrath and *Lin* [10.108] show that a model that does not include all active location effects raises the probability of falsely identifying significant dispersion factors. They show analytically that without replication it is impossible to deconfound a dispersion effect from two location effects.

Generalized Linear Model

The linear modeling discussed in this paper assumes normality and constant variance. When the data does not demonstrate these properties, the most common approach is to model a compliant, transformed response. In many cases this is hard or impossible. The general linear model (GLM) was developed by *Nelder* and *Wedderburn* [10.109] as a way of modeling data whose probability distribution is any member of the single parameter exponential family.

The GLM is fitted by obtaining the maximum likelihood estimates for the coefficients to the terms in the linear predictor, which may contain continuous, categorical, interaction and polynomial terms. *Nelder* and *Lee* [10.110] argue that the GLM can extend the class of useful models for RD experiments to data-sets wherein a simple transformation cannot necessarily satisfy the

important criteria of normality, separation and parsimony. Several examples illustrate how the link functions are chosen.

Engel and *Huele* [10.111] integrate the GLM within the RSM approach to RD. Nonconstant variance is assumed and models for process mean and variance are obtained from a heteroscedastic linear model of the conditional process response. The authors claim that nonlinear models and tolerances can also be studied with this approach. *Hamada* and *Nelder* [10.112] apply the techniques described in *Nelder* and *Lee* [10.110] to three quality improvement examples to emphasize the utility of the GLM in RD problems over its wider class of distributions.

Bayesian Modeling

Bayesian methods of analysis are steadily finding wider employment in the statistical world as useful alternatives to frequentist methods. In this section we mention several references on Bayesian modeling of the data.

Using a Bayesian GLM, *Chipman* and *Hamada* [10.113] overcome the GLM's potentially infinite likelihood estimates from categorical data taken from fractional factorial designs. *Chipman* [10.114] uses the model selection methodology of *Box* and *Meyer* [10.115] in conjunction with priors for variable selection with related predictors. For optimal choice of control factor settings he finds posterior distributions to assess the effect of model and parameter uncertainty.

10.3 Robust Design for Multiple Responses

Earlier we discussed loss and utility functions and showed how the relation between off-target and variance components underlies the loss function optimization strategies for single responses. Multi-response optimization typically combines the loss or utility functions of individual responses into a multivariate function to evaluate the sets of responses created by a particular set of design variable settings. This section is divided into two subsections which, respectively, deal with the additive and multiplicative combination of loss and utility functions, respectively.

10.3.1 Additive Combination of Univariate Loss, Utility and SNR

The majority of multiple response approaches additively combine the univariate loss or SNR performance measures discussed. In this section we review how these performance measures are additively combined and their relative advantages and disadvantages as multivariate objective functions.

Multivariate Quadratic Loss

For univariate responses, expected squared-error loss is a convenient way to evaluate the loss caused by deviation from target because of its decomposition into squared off-target and variance terms. A natural extension of this loss function to multiple correlated responses is the multivariate quadratic function of the deviation vector $(\boldsymbol{Y}-\tau)$ where $\boldsymbol{Y}=(Y_1,\ldots,Y_r)^{\mathrm{T}}$ and $\tau=(t_1,\ldots,t_r)^{\mathrm{T}}$, i. e.,

$$\mathrm{MQL}(\boldsymbol{Y},\tau)=(\boldsymbol{Y}-\tau)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{Y}-\tau)\,, \tag{10.18}$$

where $\boldsymbol{A}$ is a positive definite constant matrix. The values of the constants in $\boldsymbol{A}$ are related to the costs of nonoptimal design, such as the costs related to repairing and/or scrapping noncompliant product. In general, the diagonal elements of $\boldsymbol{A}$ represent the weights of the r characteristics and the off-diagonal elements represent the costs related to pairs of responses being simultaneously off-target.

It can be shown that, if $\boldsymbol{Y}$ follows a multivariate normal distribution with mean vector $E(\boldsymbol{Y})$ and covariance matrix $\boldsymbol{\Sigma_Y}$, the average (expected) loss can be written as:

$$\begin{aligned} E(\mathrm{MQL}) &= E(\boldsymbol{Y}-\tau)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{Y}-\tau) \\ &= \mathrm{Tr}(\boldsymbol{A}\boldsymbol{\Sigma_Y}) \\ &\quad +[E(\boldsymbol{Y})-\tau]^{\mathrm{T}}\boldsymbol{A}[E(\boldsymbol{Y})-\tau]. \end{aligned} \tag{10.19}$$

The simplest approach to solving the RD problem is to apply algorithms to directly minimize the average loss function in (10.19). Since the mean vector and covariance matrix are usually unknown, they can be estimated by the sample mean vector and sample covariance matrix or a fitted model based on a sample of observations of the multivariate responses. The off-target vector product $[E(\boldsymbol{Y})-\tau]^{\mathrm{T}}\boldsymbol{A}[E(\boldsymbol{Y})-\tau]$ and $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{\Sigma_Y})$ are multivariate analogs to the squared off-target component and variance of the univariate squared-error loss function. This decomposition shows how moving all response means to target simplifies the expected multivariate loss to the $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{\Sigma_Y})$ term. The trace-covariance term shows how the values of $\boldsymbol{A}$ and the covariance matrix $\boldsymbol{\Sigma_Y}$ directly affect the expected multivariate loss.

Optimization of Multivariate Loss Functions

For the expected multivariate quadratic loss of (10.19), *Pignatiello* [10.16] introduced a two-step procedure for finding the design variable settings that minimize this composite cost of poor quality. *Tsui* [10.18] extended Pignatiello's two-step procedure to situations where responses may be NTB, STB or LTB.

To this point we have examined squared-error loss functions whose expected value is decomposed into off-target and variance components. *Ribeiro* and *Elsayed* [10.116] introduced a multivariate loss function which additionally considers fluctuation in the supposedly fixed design variable settings. *Ribeiro* et al. [10.117] add a term for manufacturing cost to the gradient loss function of Ribeiro and Elsayed.

Additive Formation of Multivariate Utility Functions

Kumar et al. [10.118] suggest creating a multiresponse utility function as the additive combination of utility functions from the individual responses where the goal is to find the set of design variable settings that maximizes overall utility. Additional papers related to this technique include *Artiles-Leon* [10.119] and *Ames* et al. [10.120].

Quality Loss Functions for Nonnegative Variables

Joseph [10.121] argues that, in general, processes should not be optimized with respect to a single STB or LTB characteristic, rather to a combination of them. He introduces a new class of loss functions for nonnegative variables which accommodates the cases of unknown target and asymmetric loss and which can be additively combined for the multiresponse case.

10.3.2 Multivariate Utility Functions from Multiplicative Combination

In this section, a multivariate desirability function is constructed from the geometric average of the individual desirability functions of each response.

The geometric average of r components $(d_1, \ldots, d_r)$ is the rth root of their products:

$$\mathrm{GA}(d_1, \ldots, d_r) = \left(\prod_{i=1}^{r} d_i \right)^{\frac{1}{r}} . \qquad (10.20)$$

The GA is then a multiplicative combination of the individuals. When combining individual utility functions whose values are scaled between zero and one, the GA yields a value less than or equal to the lowest individual utility value. When rating the composite quality of a product, this prevents any single response from reaching an unacceptable value, since a very low value on any crucial characteristic (such as safety features or cost) will render the entire product worthless to the end user.

Modifications of the Desirability Function

In order to allow the DM to place the ideal target value anywhere within the specifications, *Derringer* and *Suich* [10.122] introduced a modified version of the desirability functions of *Harrington* [10.99] which encompassed both one-sided and two-sided response specifications. Additional extensions of the multivariate desirability function were made by *Kim* and *Lin* [10.123].

10.3.3 Alternative Performance Measures for Multiple Responses

Duffy et al. [10.124] propose using a reasonably precise estimate of multivariate yield, obtained via Beta distribution discrete point estimation, as an efficient alternative to Monte Carlo simulation. This approach is limited to independently distributed design variables. *Fogliatto* and *Albin* [10.125] propose using predictor variance as a multiresponse optimization criterion. They measure predictive variance as the coefficient of variance (CV) of prediction since it represents a normalized measure of prediction variance. *Plante* [10.126] considers the use of maximal process capability as the criterion for choosing control variable settings in multiple response RD situations. He uses the concepts of process capability and desirability to develop process capability measures for multiple response systems.

10.4 Dynamic Robust Design

10.4.1 Taguchi's Dynamic Robust Design

Up to this point, we've discussed only static RD, where the targeted response is a given, fixed level and is only affected by control and noise variables. In dynamic robust design (DRD) a third type of variable exists, the signal variable M whose magnitude directly affects the mean value of the response. The experimental design

recommended by Taguchi for DRD is the product array consisting of an inner control array crossed with an outer array consisting of the sensitivity factors and a compound noise factor.

A common choice of dynamic loss function is the quadratic loss function popularized by Taguchi,

$$L[Y, t(M)] = A_0[Y - t(M)]^2 , \tag{10.21}$$

where A_0 is a constant. This loss function provides a good approximation to many realistic loss functions. It follows that the average loss becomes

$$\begin{aligned} R(\boldsymbol{x}) &= A_0 E_M E_{N,\epsilon}[Y - t(M)]^2 \\ &= A_0 E_M \left\{ \mathrm{Var}_{N,\epsilon}(Y) + [E_{N,\epsilon}(Y) - t(M)]^2 \right\} . \end{aligned} \tag{10.22}$$

Taguchi identifies dispersion and sensitivity effects by modeling SNR respectively as a function of control factors and sensitivity factors. His two-step procedure for DRD finds control factor settings to minimize SNR and sets other, non-SNR related control variables to adjust the process to the targeted sensitivity level.

10.4.2 References on Dynamic Robust Design

Ghosh and *Derderian* [10.127] introduce the concept of robustness of the experimental plan itself to the noise factors present when conducting DRD. For combined arrays they consider blocked and split-plot designs and for product arrays they consider univariate and multivariate models. In product arrays they do this by choosing settings which minimize the noise factor effects on process variability and for the combined array they attempt to minimize the interaction effects between control and noise factors.

Wasserman [10.128] clarifies the use of the SNR for the dynamic case by explaining it in terms of linear modeling of process response. He expresses the dynamic response as a linear model consisting of a signal factor, the true sensitivity (β) at specific control variable settings, and an error term. *Miller* and *Wu* [10.129] prefer the term signal-response system to dynamic robust design for its intuitive appeal and identify two distinct types of signal-response systems. They call them measurement systems and multiple target systems, where this distinction determines the performance measure used to find the optimal control variable settings.

Lunani, *Nair* and *Wasserman* [10.130] present two new graphical procedures for identifying suitable measures of location and dispersion in RD situations with dynamic experimental designs. *McCaskey* and *Tsui* [10.131] show that Taguchi's two-step procedure for dynamic systems is only appropriate for multiplicative models and develop a procedure for dynamic systems under an additive model. For a dynamic system this equates to minimizing the sum of process variance and bias squared over the range of signal values.

Tsui [10.132] compares the effect estimates obtained using the response model approach and Taguchi's approach for dynamic robust design problems. Recent publications on DRD include *Joseph* and *Wu* [10.133], *Joseph* and *Wu* [10.134] and *Joseph* [10.135].

10.5 Applications of Robust Design

10.5.1 Manufacturing Case Studies

Mesenbrink [10.136] applied the techniques of RD to optimize three performance measurements of a high volume wave soldering process. They achieved significant quality improvement using a mixed-level fractional factorial design to collect ordered categorical data regarding the soldering quality of component leads in printed circuit boards. *Lin* and *Wen* [10.137] apply RD to improve the uniformity of a zinc coating process.

Chhajed and *Lowe* [10.138] apply the techniques of RD to the problem of structured tool management. For the cases of tool selection and tool design they use Taguchi's quadratic loss function to find the most cost effective way to accomplish the processing of a fixed number of punched holes in sheet metal products.

10.5.2 Reliability

Reliability is the study of how to make products and processes function for longer periods of time with minimal interruption. It is a natural area for RD application and the Japanese auto industry has made huge strides in this area compared to its American counterpart. In this section several authors comment on the application of RD to reliability.

Hamada [10.139] demonstrates the relevance of RD to reliability improvement. He recommends the response model approach for the additional information it provides on control–noise interactions and suggests alternative performance criteria for maximizing reliability. *Kuhn* et al. [10.140] extend the methods of *Myers* et al. [10.103] for linear models and normally

distributed data to achieve a robust process when time to an event is the response.

10.5.3 Tolerance Design

This paper has focused on RD, which is synonymous with Taguchi's methods of parameter design. Taguchi has also made significant contributions in the area of tolerance design. This section reviews articles which examine developments in the techniques of tolerance design.

D'errico and *Zaino* [10.141] propose a modification of Taguchi's approach to tolerance design based on a product Gaussian quadrature which provides better estimates of high-order moments and outperforms the basic Taguchi method in most cases. *Bisgaard* [10.142] proposes using factorial experimentation as a more scientific alternative to trial and error to design tolerance limits when mating components of assembled products.

Zhang and *Wang* [10.143] formulate the robust tolerance problem as a mixed nonlinear optimization model and solve it using a simulated annealing algorithm. The optimal solution allocates assembly and machining tolerances so as to maximize the product's insensitivity to environmental factors. *Li* and *Wu* [10.55] combined parameter design with tolerance design.

Maghsoodloo and *Li* [10.144] consider linear and quadratic loss functions for determining an optimal process mean which minimizes the expected value of the quality loss function for asymmetric tolerances of quality characteristics. *Moskowitz* et al. [10.145] develop parametric and nonparametric methods for finding economically optimal tolerance allocations for a multivariable set of performance measures based on a common set of design parameters.

References

10.1 D. C. Montgomery: *Introduction to Statistical Quality Control*, Vol. 3rd edn. (Wiley, New York 1996)

10.2 D. C. Montgomery, W. H. Woodall: A discussion on statistically-based process monitoring and control, J. Qual. Technol. **29**, 121–162 (1997)

10.3 W. H. Woodall, K.-L. Tsui, G. R. Tucker: A review of statistical and fuzzy quality control charts based on categorical data. In: *Frontiers in Statistical Quality Control*, Vol. 5, ed. by H.-J. Lenz, P. Wilrich (Physica, Heidelberg 1997) pp. 83–89

10.4 G. Taguchi: *Introduction to Quality Engineering: Designing Quality into Products and Processes* (Asian Productivity Organization, Tokyo 1986)

10.5 M. S. Phadke, R. N. Kackar, D. V. Speeney, M. J. Grieco: Off-line quality control integrated circuit fabrication using experimental design, The Bell Sys. Tech. J. **1**, 1273–1309 (1983)

10.6 R. N. Kackar, A. C. Shoemaker: Robust design: A cost effective method for improving manufacturing process, ATT Tech. J. **65**, 39–50 (1986)

10.7 W. H. Woodall, D. C. Montgomery: Research issues and ideas in statistical process control, J. Qual. Technol. **31**, 376–386 (1999)

10.8 C. A. Lowry, D. C. Montgomery: A review of multivariate control charts, IIE Trans. Qual. Reliab. **27**, 800–810 (1995)

10.9 R. L. Mason, C. W. Champ, N. D. Tracy, S. J. Wierda, J. C. Young: Assessment of multivariate process control techniques, J. Qual. Technol. **29**, 140–143 (1997)

10.10 R. N. Kackar: Off-line quality control, parameter design, and the Taguchi method, J. Qual. Technol. **17**, 176–209 (1985)

10.11 R. V. Leon, A. C. Shoemaker, R. N. Kackar: Performance measure independent of adjustment: An explanation and extension of Taguchi's signal to noise ratio, Technometrics **29**, 253–285 (1987)

10.12 G. E. P. Box: Signal to noise ratios, performance criteria and transformations, Technometrics **30**, 1–31 (1988)

10.13 V. N. Nair: Taguchi's parameter design: A panel discussion, Technometrics **34**, 127–161 (1992)

10.14 K.-L. Tsui: A critical look at Taguchi's modelling approach, J. Appl. Stat. **23**, 81–95 (1996)

10.15 N. Logothetis, A. Haigh: Characterizing and optimizing multi-response processes by the Taguchi method, Qual. Reliab. Eng. Int. **4**, 159–169 (1988)

10.16 J. J. Pignatiello: Strategies for robust multiresponse quality engineering, IIE Trans. Qual. Reliab. **25**, 5–25 (1993)

10.17 E. A. Elsayed, A. Chen: Optimal levels of process parameters for products with multiple characteristics, Int. J. Prod. Res. **31**, 1117–1132 (1993)

10.18 K.-L. Tsui: Robust design optimization for multiple characteristic problems, Int. J. Prod. Res. **37**, 433–445 (1999)

10.19 W. A. Shewhart: *Economic Control of Quality of Manufactured Product* (Van Nostrand, New York 1931)

10.20 G. E. P. Box, T. Kramer: Statistical process monitoring and feedback adjustment – A discussion, Technometrics **34**, 251–285 (1992)

10.21 W. E. Deming: *The New Economics: For Industry, Government, Education, 2nd edn.* (MIT Center for Advanced Engineering Study, Cambridge 1996)

10.22 S. W. Roberts: Control chart tests based on geometric moving averages, Technometrics **1**, 239–250 (1959)

10.23 J. M. Lucas, M. S. Saccucci: Exponentially weighted moving average control schemes: Properties and enhancements, Technometrics **32**, 1–12 (1990)

10.24 E. S. Page: Continuous inspection schemes, Biometrika **41**, 100–115 (1954)

10.25 T. J. Harris, W. M. Ross: Statistical process control for correlated observations, Cdn. J. Chem. Eng. **69**, 48–57 (1991)

10.26 L. C. Alwan: Effects of autocorrelation on control chart performance, Commun. Stat. Theory Methods **41**, 1025–1049 (1992)

10.27 R. A. Johnson, M. Bagshaw: The effect of serial correlation on the performance of CUSUM test, Technometrics **16**, 103–112 (1974)

10.28 M. Bagshaw, R. A. Johnson: The effect of serial correlation on the performance of CUSUM test II, Technometrics **17**, 73–80 (1975)

10.29 N. F. Zhang: A statistical control chart for stationary process data, Technometrics **40**, 24–38 (1998)

10.30 W. Jiang, K.-L. Tsui, W. H. Woodall: A new SPC monitoring method: The ARMA chart, Technometrics **42**, 399–410 (2000)

10.31 L. C. Alwan, H. V. Roberts: Time-series modeling for statistical process control, J. Bus. Econ. Stat. **6**, 87–95 (1988)

10.32 C. R. Superville, B. M. Adams: An evaluation of forecast-based quality control schemes, Commun. Stat. Sim. Comp. **23**, 645–661 (1994)

10.33 D. G. Wardell, H. Moskowitz, R. D. Plante: Control charts in the presence of data correlation, Man. Sci. **38**, 1084–1105 (1992)

10.34 D. G. Wardell, H. Moskowitz, R. D. Plante: Run-length distributions of special-cause control charts for correlated observations, Technometrics **36**, 3–17 (1994)

10.35 S. A. Van der Wiel: Monitoring processes that wander usingtegrated moving average models, Technometrics **38**, 139–151 (1996)

10.36 D. C. Montgomery, C. M. Mastrangelo: Some statistical process control methods for autocorrelated data, J. Qual. Technol. **23**, 179–204 (1991)

10.37 W. Jiang, H. Wu, F. Tsung, V. N. Nair, K.-L. Tsui: PID-based control charts for process monitoring, Technometrics **44**, 205–214 (2002)

10.38 D. W. Apley, J. Shi: The GLRT for statistical process control of autocorrelated processes, IIE Trans. Qual. Reliab. **31**, 1123–1134 (1999)

10.39 W. Jiang: Multivariate control charts for monitoring autocorrelated processes, J. Qual. Technol. **36**, 367–379 (2004)

10.40 D. W. Apley, F. Tsung: The autoregressive T^2 chart for monitoring univariate autocorrelated processes, J. Qual. Technol. **34**, 80–96 (2002)

10.41 G. C. Runger, T. R. Willemain: Model-based and model-free control of autocorrelated processes, J. Qual. Technol. **27**, 283–292 (1995)

10.42 G. C. Runger, T. R. Willemain: Batch means charts for autocorrelated data, IIE Trans. Qual. Reliab. **28**, 483–487 (1996)

10.43 D. P. Bischak, W. D. Kelton, S. M. Pollock: Weighted batch means for confidence intervals in steady-state simulations, Man. Sci. **39**, 1002–1019 (1993)

10.44 C. Alexopoulos, D. Goldsman, K.-L. Tsui, W. Jiang: SPC monitoring and variance estimation. In: *Frontiers in Statistical Quality Control*, Vol. 7, ed. by H.-J. Lenz, P. T. Wilrich (Physica, Heidelberg 2004) pp. 194–210

10.45 G. E. P. Box, G. M. Jenkins: *Time Series Analysis, Forecasting and Control* (Prentice-Hall, Englewood Cliffs 1976)

10.46 G. E. P. Box, A. Luceno: *Statistical Control by Monitoring and Feedback Adjustment* (Wiley, New York 1997)

10.47 S. A. Van der Wiel, W. T. Tucker, F. W. Faltin, N. Doganaksoy: Algorithmic statistical process control: Concepts and application, Technometrics **34**, 278–281 (1992)

10.48 W. T. Tucker, F. W. Faltin, S. A. Van der Wiel: Algorithmic statistical process control: An elaboration, Technometrics **35**, 363–375 (1993)

10.49 E. Sachs, A. Hu, A. Ingolfsson: Run by run process control: Combining SPC and feedback control, IEEE Trans. Semicond. Manufact. **8**, 26–43 (1995)

10.50 W. S. Messina, D. C. Montgomery, J. B. Keats, G. C. Runger: Strategies for statistical monitoring of integral control for the continuous process industry. In: *Statistical Applications in Process Control*, ed. by J. B. Keats, D. C. Montgomery (Marcel-Dekker, New York 1996) pp. 193–215

10.51 C. Capilla, A. Ferrer, R. Romero, A. Hualda: Integration of statistical and engineering process control in a continuous polymerization process, Technometrics **41**, 14–28 (1999)

10.52 W. Jiang, K.-L. Tsui: An economic model for integrated APC and SPC control charts, IIE Trans. Qual. Reliab. **32**, 505–513 (2000)

10.53 W. Jiang, K.-L. Tsui: SPC monitoring of MMSE- and PI-controlled processes, J. Qual. Technol. **34**, 384–398 (2002)

10.54 T. Kourti, P. Nomikos, J. F. MacGregor: Analysis, monitoring and fault diagnosis of batch processesusing multiblock and multiway PLS, J. Proc. Control **5**, 277–284 (1995)

10.55 W. Li, C. F. J. Wu: An integrated method of parameter design and tolerance design, Qual. Eng. **11**, 417–425 (1999)

10.56 F. Tsung, D. W. Apley: The dynamic T-squared chart for monitoring feedback-controlled processes, IIE Trans. Qual. Reliab. **34**, 1043–1054 (2002)

10.57 W. Jiang: A joint spc monitoring scheme for APC-controlled processes, IIE Trans. Qual. Reliab. , 1201–1210 (2004)

10.58 R. A. Boyles: Phase I analysis for autocorrelated processes, J. Qual. Technol. **32**, 395–409 (2000)

10.59 J. D. Cryer, T. P. Ryan: The estimation of Sigma for an X chart: MR/d2 or S/d4 ?, J. Qual. Technol. **22**, 187–192 (1990)

10.60 J. Berube, V. Nair: Exploiting the inherent structure in robust parameter design experiments, Stat. Sinica **8**, 43–66 (1998)

10.61 A. Luceno: Performance of discrete feedback adjustment schemes with dead band under stationary versus non-stationary stochastic disturbance, Technometrics **27**, 223–233 (1998)

10.62 D. W. Apley, H. C. Lee: Design of exponentially weighted moving average control charts for autocorrelated processes with model uncertainty, Technometrics **45**, 187–198 (2003)

10.63 H. Hotelling: Multivariate quality control. In: *Techniques of Statistical Analysis*, ed. by C. Eisenhart, M. W. Hastay, W. A. Wallis (McGraw-Hill, New York 1947)

10.64 C. A. Lowry, W. H. Woodall, C. W. Champ, S. E. Rigdon: A multivariate exponential weighted moving average control chart, Technometrics **34**, 46–53 (1992)

10.65 K. B. Crosier: Multivariate generalizations of cumulative sum quality control schemes, Technometrics **30**, 291–303 (1988)

10.66 J. J. Pignatiello, G. C. Runger: Comparisons of multivariate CUSUM charts, J. Qual. Technol. **22**, 173–186 (1990)

10.67 J. E. Jackson: Multivariate quality control, Commun. Stat. Theory Methods **14**, 2657–2688 (1985)

10.68 R. L. Mason, N. D. Tracy, J. C. Young: Decomposition of T^2 for multivariate control chart interpretation, J. Qual. Technol. **27**, 99–108 (1995)

10.69 D. M. Hawkins: Regression adjustment for variables in multivariate quality control, J. Qual. Control **25**, 170–182 (1993)

10.70 A. J. Hayter, K.-L. Tsui: Identification and quantification in multivariate quality control problems, J. Qual. Technol. **26**, 197–208 (1994)

10.71 W. Jiang, K.-L. Tsui: Comparison of individual multivariate control charts, submitted for publication

10.72 W. J. Welch, T.-K. Yu, S. M. Kang, J. Sacks: Computer experiments for quality control by parameter design, J. Qual. Technol. **22**, 15–22 (1990)

10.73 S. Ghosh, E. Derderian: Robust experimental plan and its role in determining robust design against noise factors, The Statistican **42**, 19–28 (1993)

10.74 A. Miller, R. R. Sitter, C. F. J. Wu, D. Long: Are large Taguchi-style experiments necesarry? A reanalysis of gear and pinion data, Qual. Eng. **6**, 21–38 (1993)

10.75 J. M. Lucas: Using response surface methodology to achieve a robust process, Annual Quality Congress Transactions, Milwaukee, WI **45**, 383–392 (1990)

10.76 P. R. Rosenbaum: Some useful compound dispersion experiments in quality design, Technometrics **38**, 248–260 (1996)

10.77 C. F. J. Wu, M. Hamada: *Experiments: Planning, Analysis and Parameter Design Optimization* (Wiley, New York 2000)

10.78 R. N. Kackar, K.-L. Tsui: Interaction graphs: graphical aids for planning experiments, J. Qual. Technol. **22**, 1–14 (1990)

10.79 R. N. Kackar, E. S. Lagergren, J. J. Filliben: Taguchi's fixed-element arrays are fractional factorials, J. Qual. Technol. **23**, 107–116 (1991)

10.80 X. Hou, C. F. J. Wu: On the determination of robust settings in parameter design experiments, Univ. Michigan Tech. Rep. **321** (1997)

10.81 D. Bingham, R. Sitter: Minimum-aberration two-level fractional factorial split-plot designs, Technometrics **41**, 62–70 (1999)

10.82 W. Dumouchel, B. Jones: A simple bayesian modification of D-optimal designs to reduce dependence on an assumed model, Technometrics **36**, 37–47 (1994)

10.83 A. C. Atkinson, R. D. Cook: D-optimum designs for heteroscedastic linear models, J. Am. Stat. Assoc. **90**, 204–212 (1994)

10.84 G. G. Vining, D. Schaub: Experimental designs for estimating both mean and variance functions, J. Qual. Technol. **28**, 135–147 (1996)

10.85 S. Chang: An algorithm to generate near D-optimal designs for multiple response surface models, IIE Trans. Qual. Reliab. **29**, 1073–1081 (1997)

10.86 D. P. Mays: Optimal central composite designs in the presence of dispersion effects, J. Qual. Technol. **31**, 398–407 (1999)

10.87 M. Pledger: Observable uncontrollable factors in parameter design, J. Qual. Technol. **28**, 153–162 (1996)

10.88 P. R. Rosenbaum: Blocking in compound dispersion experiments, Technometrics **41**, 125–134 (1999)

10.89 W. Li, C. J. Nachtsheim: Model-robust factorial designs, Technometrics **42**, 345–352 (2000)

10.90 K. L. Tsui, A. Li: Analysis of smaller-and-larger-the better robust design experiments, Int. J. Ind. Eng. **1**, 193–202 (1994)

10.91 J. Berube, C. F. J. Wu: Signal-to-noise ratio and related measures parameter design optimization, Univ. Michigan Tech. Rep. **321** (1998)

10.92 S. Maghsoodloo: The exact relation of Taguchi's signal-to-noise ratio to his quality loss function, J. Qual. Technol. **22**, 57–67 (1990)

10.93 R. V. Leon, C. F. J. Wu: Theory of performance measures in parameter design, Stat. Sinica **2**, 335–358 (1992)

10.94 V. N. Nair, D. Pregibon: A data analysis strategy for quality engineering experiments, ATT Tech. J. **65**, 73–84 (1986)

10.95 G. G. Vining, R. H. Myers: Combining Taguchi and response surface philosophies: A dual response approach, J. Qual. Technol. **22**, 38–45 (1990)

10.96 K. A. Copeland, P. R. Nelson: Dual response optimization via direct function minimization, J. Qual. Technol. **28**, 331–336 (1996)

10.97 E. Del Castillo, D. C. Montgomery: A nonlinear programming solution to the dual response problem, J. Qual. Technol. **25**, 199–204 (1993)

10.98 D. K. Lin, W. Tu: Dual response surface optimization, J. Qual. Technol. **28**, 496–498 (1995)

10.99 E. C. Harrington: The desirability function, Ind. Qual. Control **21**, 494–498 (1965)

10.100 K. K. Ng, K.-L. Tsui: Expressing variability and yield with a focus on the customer, Qual. Eng. **5**, 255–267 (1992)

10.101 V. R. Joseph, C. F. J. Wu: Operating window experiments: A novel approach to quality improvement, J. Qual. Technol. **34**, 345–354 (2002)

10.102 V. R. Joseph, C. F. J. Wu: Failure amplification method: An information maximization approach to categorical response optimization, Technometrics **46**, 1–31 (2004)

10.103 R. H. Myers, A. I. Khuri, G. Vining: Response surface alternatives to the Taguchi robust parameter design approach, Am. Stat. **46**, 131–139 (1992)

10.104 R. H. Myers, Y. Kim, K. L. Griffiths: Response surface methods and the use of noise variables, J. Qual. Technol. **29**, 429–440 (1997)

10.105 G. E. P. Box, R. D. Meyer: Dispersion effects from fractional designs, Technometrics **28**, 19–28 (1986)

10.106 R. V. Lenth: Quick and easy analysis of unreplicated factorials, Technometrics **31**, 469–473 (1989)

10.107 G. H. Pan: The impact of unidentified location effects on dispersion-effects identification from unreplicated factorial designs, Technometrics **41**, 313–326 (1999)

10.108 R. N. McGrath, D. K. Lin: Confounding of location and dispersion effects in unreplicated fractional factorials, J. Qual. Technol. **33**, 129–139 (2001)

10.109 J. A. Nelder, R. W. Wedderburn: Generalized linear models, J. Qual. Technol. **14**, 370–384 (1972)

10.110 J. A. Nelder, Y. G. Lee: Generalized linear models for the analysis of Taguchi type experiments, J. Qual. Technol. **7**, 107–120 (1991)

10.111 J. Engel, A. F. Huele: Joint modeling of mean and dispersion, J. Qual. Technol. **38**, 365–373 (1996)

10.112 M. Hamada, J. A. Nelder: Generalized linear models for quality-improvement experiments, J. Qual. Technol. **29**, 292–304 (1997)

10.113 H. Chipman, M. Hamada: Bayesian analysis of ordered categorical data from industrial experiments, Technometrics **38**, 1–10 (1996)

10.114 H. Chipman: Handling uncertainty in analysis of robust design experiments, J. Qual. Technol. **30**, 11–17 (1998)

10.115 G. E. P. Box, R. D. Meyer: Finding the active factors in fractionated screening experiments, J. Qual. Technol. **25**, 94–105 (1993)

10.116 J. L. Ribeiro, E. A. Elsayed: A case study on process optimization using the gradient loss function, Int. J. Prod. Res. **33**, 3233–3248 (1995)

10.117 J. L. Ribeiro, F. Fogliatto, C. S. ten Caten: Minimizing manufacturing and quality costs in multiresponse optimization, Qual. Eng. **13**, 191–201 (2000)

10.118 P. Kumar, P. B. Barua, J. L. Gaindhar: Quality optimization (multi-characteristics) through Taguchi's technique and utility concept, Qual. Reliab. Eng. Int. **16**, 475–485 (2000)

10.119 N. Artiles-Leon: A pragmatic approach to multiple-response problems using loss functions, Qual. Eng. **9**, 475–485 (1996)

10.120 A. E. Ames, N. Mattucci, S. MacDonald, G. Szonyi, D. M. Hawkins: Quality loss functions for optimization across multiple response surfaces, J. Qual. Technol. **29**, 339–346 (1997)

10.121 V. R. Joseph: Quality loss functions for nonnegative variables and their applications, J. Qual. Technol. **36**, 129–138 (2004)

10.122 G. Derringer, R. Suich: Simultaneous optimization of several response variables, J. Qual. Technol. **12**, 214–219 (1980)

10.123 K.-J. Kim, Dennis K. J. Lin: Simultaneous optimization of mechanical properties of steel by maximizing exponential desirability functions, Appl. Stat. **49**, 311–325 (2000)

10.124 J. Duffy, S. Q. Liu, H. Moskowitz, R. Plante, P. V. Preckel: Assessing multivariate process/product yield via discrete point approximation, IIE Trans. Qual. Reliab. **30**, 535–543 (1998)

10.125 F. S. Fogliatto, S. L. Albin: Variance of predicted response as an optimization criterion in multiresponse experiments, Qual. Eng. **12**, 523–533 (2000)

10.126 R. D. Plante: Process capability: A criterion for optimizing multiple response product and process design, IIE Trans. Qual. Reliab. **33**, 497–509 (2001)

10.127 S. Ghosh, E. Derderian: Determination of robust design against noise factors and in presence of signal factors, Commun. Stat. Sim. Comp. **24**, 309–326 (1995)

10.128 G. S. Wasserman: Parameter design with dynamic characteristics: A regression perspective, Qual. Reliab. Eng. Int. **12**, 113–117 (1996)

10.129 A. Miller, C. F. J. Wu: Parameter design for signal-response systems: A different look at Taguchi's dynamic parameter design, Stat. Sci. **11**, 122–136 (1996)

10.130 M. Lunani, V. N. Nair, G. S. Wasserman: Graphical methods for robust design with dynamic characteristics, J. Qual. Technol. **29**, 327–338 (1997)

10.131 S. D. McCaskey, K.-L. Tsui: Analysis of dynamic robust design experiments, Int. J. Prod. Res. **35**, 1561–1574 (1997)
10.132 K.-L. Tsui: Modeling and analysis of dynamic robust design experiments, IIE Trans. Qual. Reliab. **31**, 1113–1122 (1999)
10.133 V. R. Joseph, C. F. J. Wu: Robust parameter design of multiple target systems, Technometrics **44**, 338–346 (2002)
10.134 V. R. Joseph, C. F. J. Wu: Performance measures in dynamic parameter design, J. Jpn. Qual. Eng. Soc. **10**, 82–86 (2002)
10.135 V. R. Joseph: Robust parameter design with feed-forward control, Technometrics **45**, 284–292 (2003)
10.136 P. Mesenbrink, J. C. Lu, R. McKenzie, J. Taheri: Characterization and optimization of a wave-soldering process, J. Am. Stat. Ass. **89**, 1209–1217 (1994)
10.137 S. M. Lin, T. C. Wen: Experimental strategy – application of Taguch's quality engineering method to zinc phosphate coating uniformity, Plat. Surf. Finishing **81**, 59–64 (1994)
10.138 D. Chhajed, T. J. Lowe: Tooling choices and product performance, IIE Trans. Qual. Reliab. **32**, 49–57 (2000)
10.139 M. Hamada: Reliability improvement via Taguchi's robust design, Qual. Reliab. Eng. Int. **9**, 7–13 (1993)
10.140 A. M. Kuhn, W. H. Carter, R. H. Myers: Incorporating noise factors into experiments with censored data, Technometrics **42**, 376–383 (2000)
10.141 J. R. D'errico, N. A. Zaino: Statistical tolerancing using a modification of Taguchi's method, Technometrics **30**, 397–405 (1988)
10.142 S. Bisgaard: Designing experiments for tolerancing assembled products, Technometrics **39**, 142–152 (1997)
10.143 C. C. Zhang, H. P. Wang: Robust design of assembly and machining tolerance allocations, IIE Trans. Qual. Reliab. **30**, 17–29 (1998)
10.144 S. Maghsoodloo, M. H. Li: Optimal asymmetric tolerance design, IIE Trans. Qual. Reliab. **32**, 1127–1137 (2000)
10.145 H. Moskowitz, R. Plante, J. Duffy: Multivariate tolerance design using quality loss, IIE Trans. Qual. Reliab. **33**, 437–448 (2001)

11. Statistical Methods for Product and Process Improvement

The first part of this chapter describes a process model and the importance of product and process improvement in industry. Six Sigma methodology is introduced as one of most successful integrated statistical tool.

Then the second section describes the basic ideas for Six Sigma methodology and the (D)MAIC(T) process for better understanding of this integrated process improvement methodology.

In the third section, "Product Specification Optimization", optimization models are developed to determine optimal specifications that minimize the total cost to both the producer and the consumer, based on the present technology and the existing process capability. The total cost consists of expected quality loss due to the variability to the consumer, and the scrap or rework cost and inspection or measurement cost to the producer. We set up the specifications and use them as a counter measure for the inspection or product disposition, only if it reduces the total cost compared with the expected quality loss without inspection. Several models are presented for various process distributions and quality loss functions.

The fourth part, "Process Optimization", demonstrates that the process can be improved during the design phase by reducing the bias or variance of the system output, that is, by changing the mean and variance of the quality characteristic of the output. Statistical methods for process optimization, such as experimental design, response surface methods, and Chebyshev's orthogonal polynomials are reviewed. Then the integrated optimization models are developed to minimize the total cost to the system of producers and customers by determining the means and variances of the controllable factors. Finally, a short summary is given to conclude this chapter.

Improving manufacturing or service processes is very important for a business to stay competitive in today's marketplace. Companies have been forced to improve their business processes because customers are always demanding better products and services. During the last 20 years, industrial organizations have become more and more interested in process improvement. Statistical methods contribute much to this activity, including design of experiments, regression analysis, response surface methodology, and their integration with optimization methods.

A process is a collection of activities that takes one or more kinds of inputs and creates a set of outputs that are of value to the customer. Everyone may be involved in various processes in their daily life, for example, ordering books from an Internet retailer, checking out in a grocery store, remodeling a home, or developing new products. A process can be graphed as shown in Fig. 11.1. The purpose of this model is to define the supplier, process inputs, the process, associated outputs, and the customer. The loops for the feedback information for continuous improvement are also shown.

As mentioned above, a process consists of many input variables and one or multiple output variables. The input variables include both controllable and uncontrollable or noise factors. For instance, for an electric circuit designed to obtain a target output voltage, the designer can specify the nominal values of resistors or capacitor, but he cannot control the variability of resistors or capacitors at any point in time or over the life cycle of the product. A typical process with one output variable is given in Fig. 11.2, where $X_1, X_2, \ldots, X_n$ are controllable variables and y is the realization of the random output variable Y.

Many companies have implemented continuous process improvement with Six Sigma methodology, such as Motorola [11.1] and GE [11.2]. Six Sigma is a customer-focused, data-driven, and robust methodology that is well rooted in mathematics and statistics. A typical process for Six Sigma quality improvement has six phases: define, measure, analyze, improve, control, and technology transfer, denoted by (D)MAIC(T). The section "Six Sigma Methodology and the (D)MAIC(T) Process" introduces the basic ideas behind Six Sigma methodology and the (D)MAIC(T) process for a better understanding of this integrated process-improvement methodology.

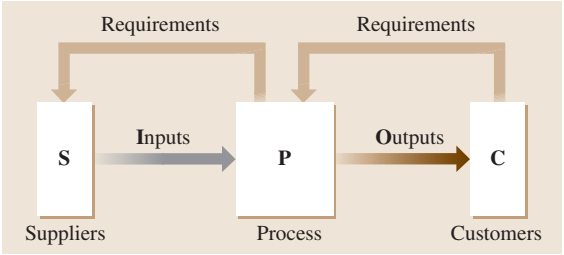


Fig. 11.1 Process model

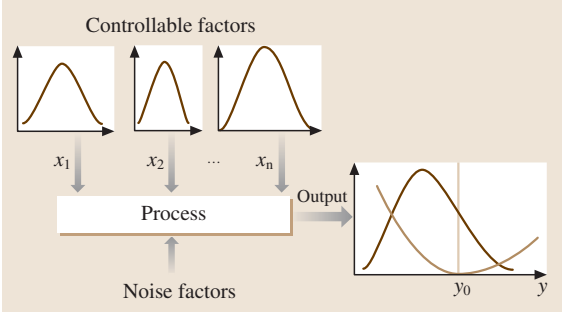


Fig. 11.2 General process with one output variable

In the section "Product Specification Optimization," we create optimization models to develop specifications that minimize the total cost to both the producer and the consumer, based on present technology and existing process capabilities. The total cost consists of expected quality loss due to the variability to the consumer and the scrap or rework cost and inspection or measurement cost to the producer. We set up the specifications and use them as a countermeasure for inspection or product disposition only if it reduces the total cost compared with the expected quality loss without inspection. Several models are presented for various process distributions and quality-loss functions.

In the section "Process Optimization," we assume that the process can be improved during the design phase by reducing the bias or variance of the system output, that is, by changing the mean and variance of the quality characteristic of the output. Statistical methods for process optimization, such as experimental design, response surface methods, and Chebyshev's orthogonal polynomials, are reviewed. Then the integrated optimization models are developed to minimize the total cost to the system of producers and customers by determining the means and variances of the controllable factors.

11.1 Six Sigma Methodology and the (D)MAIC(T) Process

The traditional evaluation of quality is based on average measures of a process/product. But customers judge the quality of process/product not only on the average, but also by the variance in each transaction or use of the product. Customers value consistent, predictable processes that deliver best-in-class levels of quality. This is what Six Sigma process strives to produce. Six Sigma methodology focuses first on reducing process variation and thus on improving the process capability.

The typical definition of a process capability index, C_{pk}, is $C_{pk} = \min((\mathrm{USL} - \hat{\mu})/(3\hat{\sigma}), (\hat{\mu} - \mathrm{LSL})/(3\hat{\sigma}))$, where USL is the upper specification limit, LSL is the lower specification limit, $\hat{\mu}$ is the point estimator of the mean, and $\hat{\sigma}$ is the point estimator of the standard deviation. If the process is centered at the middle of the specifications, which is also interpreted as the target value, i.e., $\hat{\mu} = (\mathrm{USL} + \mathrm{LSL})/(2) = y_0$, then the Six Sigma process means that $C_{pk} = 2$. In the literature, it is typically mentioned that the Six Sigma process results in 3.4 defects per million opportunities (DPMO). For this statement, we assume that the process shifts by 1.5σ over time from the target (which is assumed to be the middle point of the specifications). It implies that the realized C_{pk} is 1.5 for the Six Sigma process over time. Thus, it is obvious that 6σ requirements or C_{pk} of 1.5 is not the goal; the ideal objective is to continuously improve the process based on some economic or other higher-level objectives for the system.

At the strategic level, the goal of Six Sigma is to align an organization to its marketplace and deliver real improvement to the bottom line. At the operational level, Six Sigma strives to move product or process characteristics within the specifications required by customers, shrink process variation to the six sigma level, and reduce the cause of defects that negatively affect quality [11.3].

Six Sigma continuous improvement is a rigorous, data-driven, decision-making approach to analyzing the root causes of problems and improve the process capability to the six sigma level. It utilizes a systematic six-phase, problem-solving process called (D)MAIC(T): define, measure, analyze, improve, control, and technology transfer. Traditionally, a four-step process, MAIC, is often referred to as a general process for Six Sigma process improvement in the literature. We extend it to the six-step process, (D)MAIC(T). We want to emphasize the importance of the define (D) phase as the first phase for the problem definition and project selection, and we want to highlight technology transfer (T) as the never-ending phase for continuous applications of Six Sigma technology to other parts of the organization. The process of (D)MAIC(T) stays on track by establishing deliverables for each phase, by creating engineering models over time to reduce process variation, and by continuously improving the predictability of system performance. Each of the six phases in the (D)MAIC(T) process is critical to achieving success.

11.1.1 Define: What Problem Needs to Be Solved?

It is important to define the scope, expectations, resources, and timelines for the selected project. The definition phase for the Six Sigma approach identifies the specific scope of the project, defines the customer and critical-to-quality (CTQ) issues from the viewpoint of the customer, and develops the core processes.

11.1.2 Measure: What Is the Current Capability of the Process?

Design for Six Sigma is a data-driven approach that requires quantifying and benchmarking the process using actual data. In this phase, the performance or process capability of the process for the CTQ characteristics are evaluated.

11.1.3 Analyze: What Are the Root Causes of Process Variability?

Once the project is understood and the baseline performance documented, it is time to do an analysis of the process. In this phase, the Six Sigma approach applies statistical tools to determine the root causes of problems. The objective is to understand the process at a level sufficient to be able to formulate options for improvement. We should be able to compare the various options with each other to determine the most promising alternatives. In general, during the process of analysis, we analyze the data collected and use process maps to determine root causes of defects and prioritize opportunities for improvement.

11.1.4 Improve: Improving the Process Capability

During the improvement phase of the Six Sigma approach, ideas and solutions are incorporated to initialize

the change. Based on the root causes discovered and validated for the existing opportunity, the target process is improved by designing creative solutions to fix and prevent problems. Some experiments and trials may be implemented in order to find the best solution. If a mathematical model is developed, then optimization methods are utilized to determine the optimum solution.

11.1.5 Control: What Controls Can Be Put in Place to Sustain the Improvement?

The key to the overall success of the Six Sigma methodology is its sustainability, which seeks to make everything incrementally better on a continuous basis. The sum of all these incremental improvements can be quite large. Without continuous sustenance, over time things will get worse until finally it is time for another attempt at improvement. As part of the Six Sigma approach, performance-tracking mechanisms and measurements are put in place to assure that the gains made in the project are not lost over time and the process remains on the new course.

11.1.6 Technology Transfer: Where Else Can These Improvements Be Applied?

Ideas and knowledge developed in one part of an organization can be transferred to other parts of the organization. In addition, the methods and solutions developed for one product or process can be applied to other similar products or processes. Numbering by infinity, we keep on transferring technology, which is a never-ending phase for achieving Six Sigma quality. With technology transfer, the Six Sigma approach starts to create phenomenal returns.

There are many optimization problems in the six phases of this methodology. In the following sections, several statistical methods and optimization models are reviewed or developed to improve the quality of product or process to the six sigma level, utilizing the tools of probabilistic design, robust design, design of experiments, multivariable optimization, and simulation techniques. The goal is to investigate and explore the engineering, mathematical, and statistical bases of (D)MAIC(T) process.

11.2 Product Specification Optimization

For any process, strategic decisions have to be made in terms of the disposition of the output of the process, which may be some form of inspection or other countermeasures such as scrapping or reworking the output product. We may do zero inspection, 100% inspection, or use sampling inspection. Some of the problems with acceptance sampling were articulated by *Deming* [11.4], who pointed out that this procedure, while minimizing the inspection cost, does not minimize the total cost to the producer. *Orsini* [11.5] in her doctoral thesis explained how this results in a process of suboptimization.

Deming's inspection criterion indicates that inspection should be performed either 100% or not at all, depending on the total cost to the producer, which includes the cost of inspection, k_1, and the detrimental cost of letting a nonconforming item go further down into production, k_2. The criterion involves k_1, k_2, and p, the proportion of incoming nonconforming items. The break-even point is given by $k_1/k_2 = p$. If $k_1/k_2 < p$, then 100% inspection is called for; if $k_1/k_2 > p$, then no inspection is done under the assumption that the process is in a state of statistical control. The practicality and usefulness of Deming's criterion for a manufacturing company was illustrated by *Papadakis* [11.6], who formulated models to decide if we should do either 100% inspection or zero inspection based on the total cost to the producer.

Deming [11.4] also concludes that k_1 and k_2 are not the only costs to consider. As manufacturers try hard to meet or exceed customer expectations, the cost to the customers should be considered when planning for the inspection strategy. To meet the requirements of the current competitive global markets, we consider the cost to both consumers and producers, thus the total cost to the whole system in the general inspection model. If we decide to do 100% inspection, we should also know what specification limits are for the purpose of inspection, so that we can make decisions about the disposition of the output. The work done by Deming and others does not explicitly consider the specification limits for inspection and how to determine them.

In the following discussion, several economic models are proposed that not only explain when to do 100% inspection but also develop the specifications for the inspection. A general optimization model is developed to minimize the total cost to the system, including both the producer and the customer, utilizing the quality loss function based on some of the contribution of *Taguchi*'s

work [11.7, 8]. In particular, the optimization models with the symmetric and asymmetric quadratic quality loss function are presented to determine the optimal process mean and specification limits for inspection.

11.2.1 Quality Loss Function

The traditional concept of conformance to specifications is a binary evaluation system (Fig. 11.3). Units that meet the specification limits are labeled "good" or "conforming," and units out of specification limits are "bad" or "nonconforming." In the traditional quality concept, quality evaluation systems focus only on the nonconforming units and cost of quality is defined as cost of nonconformance. We can easily recognize the simplicity of this binary (go/no go) evaluation system, as the quality may not differ very much between a "good" item that is just within specifications and a "bad" item that is just outside specifications.

A better evaluation system should measure the quality of all the items, both within and outside specifications. As shown in Fig. 11.4, the concept of quality loss function provides a quantitative evaluation of loss caused by functional variation. We describe the derivation of the quadratic quality loss function in what follows.

Let $L_1(y)$ be a measure of losses, disutility, failure rate, or degradation associated with the quality characteristic y. $L_1(y)$ is a differentiable function in the neighborhood of the target y_0. Using Taylor's series expansion, we have

$$L_1(y) = L_1(y_0) + L_1'(y_0)(y - y_0) + L_1''(y_0)\frac{(y-y_0)^2}{2!} + \cdots .$$

The minimum quality loss should be obtained at y_0, and hence $L_1'(y_0) = 0$. Since $L_1(y_0)$ is a constant quality loss at y_0, we define the deviation loss of y from y_0 as

$$L(y) = L_1(y) - L_1(y_0) = L_1''(y_0)\frac{(y-y_0)^2}{2!} + \cdots .$$

By ignoring the higher-order terms, $L(y)$ can be approximated using a quadratic function:

$$L(y) \approx k(y - y_0)^2 ,$$

where

$$k = \frac{L_1''(y_0)}{2} .$$

If the actual quality loss function $L(y)$ is known, we should use it instead of the approximated loss function.

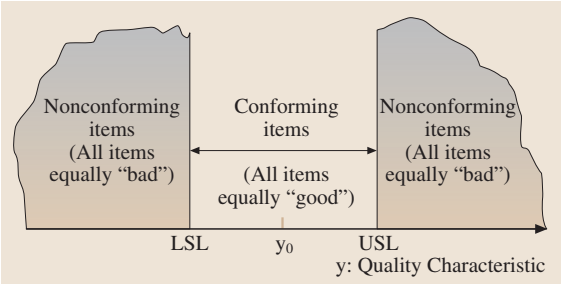


Fig. 11.3 Conformance to specifications concept of quality

Let $f(y)$ be the probability density function (pdf) of the random variable Y; then the expected loss for any given $L(y)$ is

$$\mathcal{L} = E[L(y)] = \int_{\text{all } y} L(y) f(y)\,\mathrm{d}y .$$

From this equation we can see that the expected loss depends heavily on the distribution of Y. To reduce the expected quality loss, we need to improve the distribution of Y, not just reduce the number of items outside specification limits. It is quite different from the traditional evaluation policy, which only measures the cost incurred by nonconforming quality characteristics. In the following sections, different quality loss functions are discussed for different types of quality characteristics.

"The Smaller the Better" Quality Characteristics

The objective is to reduce the value of the quality characteristic. Usually the smallest possible value for such characteristics is zero, and thus $y_0 = 0$ is the "ideal" or target value, as shown in Fig. 11.5. Some examples are wear, degradation, deterioration, shrinkage, noise

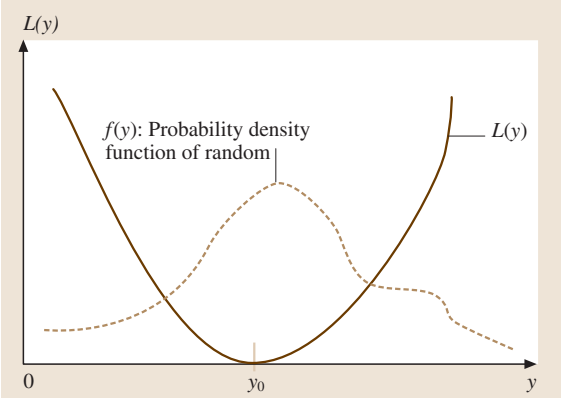


Fig. 11.4 Quality loss function $L(y)$

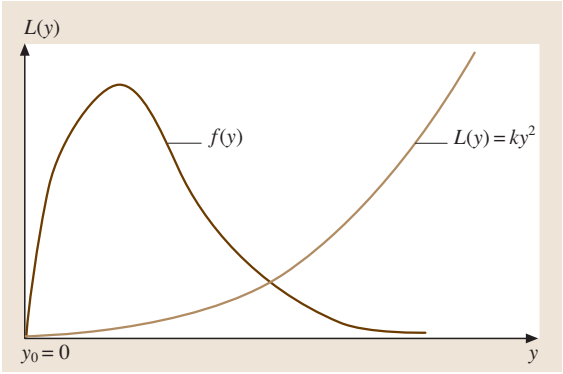


Fig. 11.5 "The smaller the better" quality characteristics

level, harmful effects, level of pollutants, etc. For such characteristics, engineers generally have an upper specification limit (USL). A good approximation of $L(y)$ is $L(y) = ky^2$, $y \geq 0$.

The expected quality loss is calculated as

$$\begin{aligned}\mathcal{L} &= E[L(y)] \\ &= \int_{\text{all } y} L(y) f(y)\,\mathrm{d}y \\ &= \int_{\text{all } y} ky^2 f(y)\,\mathrm{d}y \\ &= \int_{\text{all } y} k\left[(y-\mu)^2 + 2(y-\mu)\mu + \mu^2\right] f(y)\,\mathrm{d}y \\ &= k\left(\sigma^2 + \mu^2\right) .\end{aligned}$$

To reduce the loss, we must reduce the mean μ and the variance σ^2 simultaneously.

"The Larger the Better" Quality Characteristics

For such quality characteristics, we want to increase their value as much as possible (within a given frame of reference), as shown in Fig. 11.6. Some examples are strength, life of a system (a measure of reliability), fuel efficiency, etc. An ideal value may be infinity, though impossible to achieve. For such characteristics, engineers generally have a lower specification limit (LSL). A good approximation of $L(y)$ is

$$L(y) = \frac{k}{y^2}, \quad y \geq 0 .$$

The expected quality loss is given by

$$\mathcal{L} = E[L(y)] = \int_{\text{all } y} L(y) f(y)\,\mathrm{d}y = \int_{\text{all } y} \frac{k}{y^2} f(y)\,\mathrm{d}y .$$

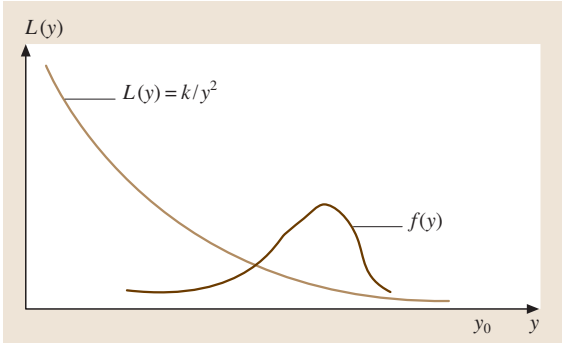


Fig. 11.6 "The larger the better" quality characteristics

Using Taylor's series expansion for $1/y^2$ around μ, we have

$$\begin{aligned}\frac{1}{y^2} &= \mu^{-2} \\ &\quad + \left(-2y^{-3}\,|_{\mu}\right)(y-\mu) \\ &\quad + 6y^{-4}\,|_{\mu}\,\frac{(y-\mu)^2}{2!} \\ &\quad + \cdots .\end{aligned}$$

By ignoring higher-order terms, we have

$$\frac{1}{y^2} \approx \frac{1}{\mu^2} + \frac{2}{\mu^3}(y-\mu) + \frac{3}{\mu^4}(y-\mu)^2 .$$

Finally, we have

$$\begin{aligned}E[L(y)] &\approx k \int_{\text{all } y} \left[\frac{1}{\mu^2} + \frac{2}{\mu^3}(y-\mu) \right. \\ &\qquad \left. + \frac{3}{\mu^4}(y-\mu)^2\right] f(y)\,\mathrm{d}y \\ &\approx k\left(\frac{1}{\mu^2} + \frac{3\sigma^2}{\mu^4}\right) .\end{aligned}$$

To reduce quality losses for the "larger the better" quality characteristics, we must increase the mean μ and reduce the variance σ^2 of Y simultaneously.

"Nominal the Best" Quality Characteristics

For such quality characteristics, we have an ideal or nominal value, as shown in Fig. 11.7. The performance of the product deteriorates as we move from each side of the nominal value. Some examples are dimensional characteristics, voltage, viscosity of a fluid, shift pressure, clearance, and so on. For such characteristics, engineers generally have both LSL and USL. An approximation of quality loss function for "nominal the best" quality characteristics is $L(y) = k(y - y_0)^2$.

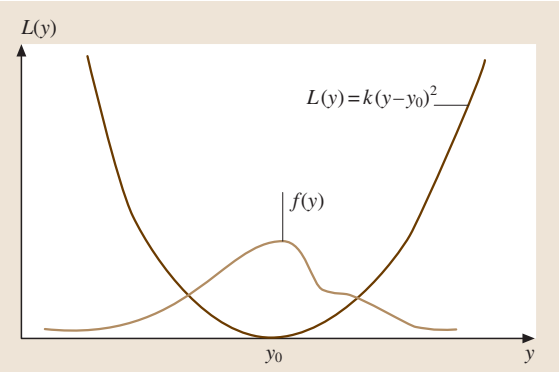


Fig. 11.7 "Nominal the best" quality characteristics

The expected quality loss is calculated as

$$\mathcal{L} = E[L(y)] = \int_{\text{all } y} L(y) f(y)\,\mathrm{d}y$$
$$= \int_{\text{all } y} k(y-y_0)^2 f(y)\,\mathrm{d}y$$
$$= k\left[\sigma^2 + (\mu - y_0)^2\right] .$$

Given the constant k, we must reduce bias $|\mu - y_0|$ and variance σ^2 to reduce the losses.

11.2.2 General Product Specification Optimization Model

Quality loss relates to cost or "loss" in dollars, not just to the manufacturer at the time of production, but also to the next consumer. The intangible losses (customer dissatisfaction, loss of customer loyalty, and eventual loss of market share), along with other tangible losses (rework, defects, down time, etc.), make up some of the components of the quality loss. Quality loss function is a way to measure losses due to variability from the target values and transform them to economic values. The greater the deviation from target, the greater the economic loss.

Variability means some kind of waste, but it is impossible to have zero variability. The common response has been to set not only a target level for performance but also a range of tolerance about that target, or specification limits, which represents "acceptable" performance. Thus if a quality characteristic falls anywhere within the specifications, it is regarded as acceptable, while if it falls outside that specifications, it is not acceptable. If the inspection has to be done to decide what is acceptable, we must know the specification limits. We consider the specifications not just from the viewpoint of the customer or the producer but from the viewpoint of the whole system. The issue is not only to decide when to do inspection, but also to decide what specifications will be applied for the inspection.

Suppose a process has been improved to its optimal capability using the present technology; then we consider the following two questions:

Question 1: Should we perform 100% inspection or zero inspection before shipping the output to the next or downstream customers?

Question 2: If 100% inspection is to be performed, how do we determine the optimal specification limits that minimize the total cost to the system, which includes both producers and consumers?

To answer the above two questions, the decision maker has to choose between the following two decisions:

Decision 1: No inspection is done, and thus we ship the whole distribution of the output to the next customer. One economic interpretation of cost to the downstream customers is the expected quality loss.

Decision 2: Do 100% inspection. It is clear that we will do the inspection and truncate the tails of the distribution only if it reduces total cost to both the producer and the consumer. If we have some arbitrary specification limits, we may very well increase the total cost by doing inspection. When we truncate the distribution by using certain specification limits, some additional costs will be incurred, such as the measurement or inspection cost (to evaluate if units meet the specifications), the rework cost, and the scrap cost. The general optimization model is

$$\text{Minimize ETC} = \text{EQL} + \text{ESC} + \text{IC} ,$$

where

ETC = Expected total cost per produced unit

EQL = Expected quality loss per unit

ESC = Expected scrap cost per unit

IC = Inspection cost per unit

and where the specification limits are the decision variables in the optimization model. Based on this general optimization model, models have been formulated under the following assumptions:

- The nature of the quality characteristics:
  - "The smaller the better"
  - "The larger the better"
  - "Target the best"

- The nature of the underlying distributions of the output:
  - Normal distribution
  - Lognormal distribution
  - Weibull distribution
- The relationship between the process mean and the target value:
  - The process mean is centered at the target: $\mu = y_0$
  - The process mean is not centered at the target: $\mu \neq y_0$
- The shape of the quality loss function:
  - Symmetric
  - Asymmetric
- The number of quality characteristics:
  - Single quality characteristic
  - Multiple quality characteristics

Kapur [11.9], *Kapur* and *Wang* [11.10], *Kapur* and *Cho* [11.11], and *Kapur* and *Cho* [11.12] have developed several models for various quality characteristics and illustrated the models with several numerical problems. *Kapur* and *Wang* [11.10] and *Kapur* [11.13] considered the normal distribution for the "target the best" single quality characteristic to develop the specification limits based on the symmetric quality loss function and also used the lognormal distribution to develop the model for the "smaller the better" single quality characteristic. For the "smaller or larger the better" single quality characteristic, *Kapur* and *Cho* [11.11] used the Weibull distribution to approximate the underlying skewed distribution of the process, because a Weibull distribution can model various shapes of the distribution by changing the shape parameter β. *Kapur* and *Cho* [11.12] proposed an optimization model for multiple quality characteristics with the multivariate normal distribution based on the multivariate quality loss function.

In the next two subsections, two optimization models are described to determine the optimal specification limits. The first model is developed for a normal distributed quality characteristic with a symmetric quality loss function, published by *Kapur* and *Wang* [11.10] and *Kapur* [11.13]. The second model is formulated for a normal distributed quality characteristic with an asymmetric quality loss function, proposed by *Kapur* and *Feng* [11.14].

11.2.3 Optimization Model with Symmetric Loss Function

We summarize the basic assumptions presented in *Kapur* and *Wang* [11.10] and *Kapur* [11.13] as below:

- The single quality characteristic is "target the best," and the target is y_0.
- The process follows a normal distribution: $Y \propto N(\mu, \sigma^2)$.
- The process mean is centered at the target: $\mu = y_0$.
- The quality loss function is symmetric about the target y_0 and given as $L(y) = k(y - y_0)^2$.

Based on these assumptions, the expected quality loss without inspection is calculated as:

$$\begin{aligned}\mathcal{L} = E[L(Y)] &= \int_{-\infty}^{\infty} k(y - y_0)^2 f(y)\,\mathrm{d}y \\ &= k\left\{[E(Y) - y_0]^2 + Var(Y)\right\} \\ &= k\left[\sigma^2 + (y_0 - \mu)^2\right] .\end{aligned}$$

After setting the process mean at the target, $\mu = y_0$, the expected loss only has the variance term, which is $\mathcal{L} = k\sigma^2$.

If we do 100% inspection, we will truncate the tails of the distribution at specification limits, which should be symmetric about the target:

$$\begin{aligned}\mathrm{LSL} &= \mu - n\sigma , \\ \mathrm{USL} &= \mu + n\sigma .\end{aligned}$$

In order to optimize the model, we need to determine the variance of the truncated normal distribution (the distribution of the units shipped to the customer), which is $\mathrm{V}(Y_\mathrm{T})$. Let $f_\mathrm{T}(y_\mathrm{T})$ be the probability density function for the truncated random variable Y_T; then we have

$$f_\mathrm{T}(y_\mathrm{T}) = \frac{1}{q} f(y_\mathrm{T}) = \frac{1}{q\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(y_\mathrm{T}-\mu)^2}{2\sigma^2}} ,$$

where

$$\begin{aligned}q &= 2\Phi(n) - 1 \\ &= \text{fraction of units shipped to customers} \\ &\quad \text{or area under normal distribution within} \\ &\quad \text{specification limits}\end{aligned}$$

and

$$\mu - n\sigma \leq y_\mathrm{T} \leq \mu + n\sigma ,$$

where $\phi(\cdot)$ is the pdf for the standard normal variable and $\Phi(\cdot)$ is the cdf for the standard normal variable. From the probability density function (pdf) we can derive the mean and variance of the truncated normal distribution as

$$E(Y_T) = \mu \,,$$
$$V(Y_T) = \sigma^2 \left[1 - \frac{2n}{2\Phi(n)-1}\phi(n)\right] .$$

It is clear that the quantity of $V(Y_T)$ is less than σ^2, which means that we reduce the variance of units shipped to the customer (Y_T). Then the expected quality loss, $\mathcal{L}_T$, for the truncated distribution is

$$\mathcal{L}_T = k\left\{[E(Y_T) - y_0]^2 + V(Y_T)\right\} = kV(Y_T) \,.$$

Then the expected quality loss per unit EQL is $q\mathcal{L}_T$, because the fraction of units shipped to customers is q. Given k, SC, and IC, we have the optimization model with only one decision variable n as

$$\begin{aligned} &\text{Minimize ETC} = q\mathcal{L}_T + (1-q)\text{SC} + \text{IC} \,, \\ &\text{subject to } \mathcal{L}_T = k\sigma^2\left[1 - \frac{2n}{2\Phi(n)-1}\phi(n)\right] , \\ &\qquad q = 2\Phi(n) - 1 \,, \\ &\qquad n \geq 0 \,. \end{aligned}$$

The above objective function is unimodal and differentiable, and hence the optimal solution can be found by differentiating the objective function with respect to n and setting it equal to zero. Thus we solve $(\partial \text{ETC}/\partial n) = 0$, and the solution is $n^* = \sqrt{\text{SC}/(k\sigma^2)}$.

Let us now consider an example for a normal process with $\mu = 10$, $\sigma = 0.50$, $y_0 = 10$, $k = 5$, IC = \$0.10, and SC = \$2.00.

Decision 1: If we do not conduct any inspection, the total expected quality loss per unit is calculated as $\text{TC} = \mathcal{L} = k\sigma^2 = 5 \times 0.50^2 = \$1.25$.

Decision 2: Let us determine the specification limits that will minimize the total expected cost by using the following optimization model:

$$\begin{aligned} &\text{Minimize ETC} = q\mathcal{L}_T + (1-q)\text{SC} + \text{IC} \\ &\qquad = 5 \times 0.5^2\,[2\Phi(n) - 1 - 2n\phi(n)] \\ &\qquad\quad + [2 - 2\Phi(n)] \times 2.00 + 0.10 \\ &\text{subject to } n \geq 0 \,. \end{aligned}$$

The optimal solution is given by $n^* = \sqrt{\text{SC}/(k\sigma^2)} = 1.26$, and $\text{ETC}^* = \$0.94 < \$1.25$. Thus, the optimal strategy is to have LSL = 9.37 and USL = 10.63, and do 100% inspection to screen the nonconforming units. The above model presents a way to develop optimum specification limits by minimizing the total cost. Also, Fig. 11.8 gives the relationship between the expected total cost ETC and n, where we can easily observe that the minimum is when $n = 1.26$.

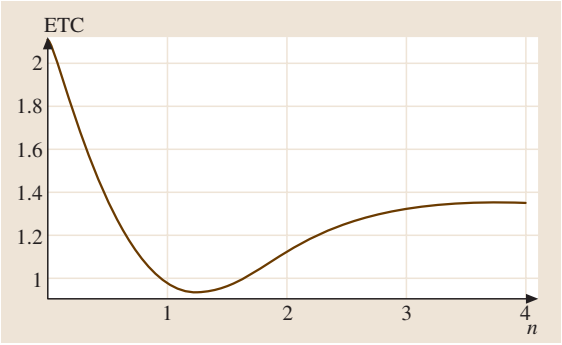


Fig. 11.8 Expected total cost vs. n

In addition to the above model for the "target the best" quality characteristic, *Kapur* and *Wang* [11.10] used the lognormal distribution to develop a model for the "smaller the better" quality characteristic. For the "smaller or larger the better" quality characteristic, *Kapur* and *Cho* [11.11] used the Weibull distribution to approximate the underlying skewed distribution of the process because a Weibull distribution can model various shapes of the distribution by changing the shape parameter β.

11.2.4 Optimization Model with Asymmetric Loss Function

The following assumptions are presented to formulate this optimization model [11.14]:

- The single quality characteristic is "target the best," and the target is y_0.
- The process follows a normal distribution: $Y \approx N(\mu, \sigma^2)$, and the probability density function of Y is $f(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$.
- The mean of the process can be easily adjusted, but the variance is given based on the present technology or the inherent capability of the process.
- The process mean may not be centered at the target: $\mu \neq y_0$, which is a possible consequence of an asymmetric loss function.
- The quality loss function is asymmetric about the target y_0, which means that the performance of the product deteriorates in the different ways as the

quality characteristic deviates to either side of the target value. An asymmetric quality loss function is given as:

$$\begin{cases} k_1(y-y_0)^2, & y \le y_0\,, \\ k_2(y-y_0)^2, & y > y_0\,. \end{cases}$$

Based on these assumptions, if we ship the whole distribution of the output to the next customer as for Decision 1, the total cost is just the expected quality loss to the customer. We can prove that the expected quality loss without truncating the distribution is:

$$\begin{aligned} \mathrm{ETC}_1 = \mathcal{L} = & \int_{-\infty}^{y_0} k_1(y-y_0)^2 f(y)\,\mathrm{d}y \\ & + \int_{y_0}^{\infty} k_2(y-y_0)^2 f(y)\,\mathrm{d}y \\ = & (k_1-k_2)\sigma(y_0-\mu)\phi\left(\frac{y_0-\mu}{\sigma}\right) \\ & + \left[\sigma^2+(y_0-\mu)^2\right] \\ & \times\left[(k_1-k_2)\Phi\left(\frac{y_0-\mu}{\sigma}\right)+k_2\right], \end{aligned}$$

where $\phi(\cdot)$ is the pdf for the standard normal variable and $\Phi(\cdot)$ is the cdf for the standard normal variable. Given k_1, k_2, and y_0 and the standard deviation σ, the total cost or the expected quality loss to the customer in this case should be minimized by finding the optimal process mean μ^*. The optimization model for Decision 1 is:

$$\begin{aligned} \text{Minimize } \mathrm{ETC}_1 = & (k_1-k_2)\sigma(y_0-\mu)\phi\left(\frac{y_0-\mu}{\sigma}\right) \\ & + \left[\sigma^2+(y_0-\mu)^2\right] \\ & \times\left[(k_1-k_2)\Phi\left(\frac{y_0-\mu}{\sigma}\right)+k_2\right] \\ \text{subject to } & \mu \in \mathbb{R}\,. \end{aligned}$$

Given k_1, k_2, y_0, and σ, ETC_1 or $\mathcal{L}$ is a convex differential function of μ, because the second derivative $\frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}\mu^2} > 0$. We know that a convex differential function obtains its global minimum at $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mu} = 0$, which is given by

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mu} = & 2(k_2-k_1)\left[\sigma^2 f(y_0)+(\mu-y_0)\int_{y_0}^{\mu} f(y)\,\mathrm{d}y\right] \\ & + (k_1+k_2)(\mu-y_0) \\ = & 0\,. \end{aligned} \tag{11.1}$$

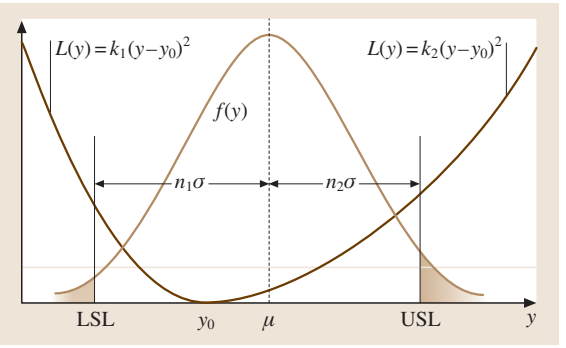


Fig. 11.9 Optimization model with asymmetric loss function

Thus, the optimal value of the process mean μ^* is obtained by solving the above equation of μ. Since the root of (11.1) cannot be found explicitly, we can use Newton's method to search the numerical solution by Mathematica.

If we do the 100% inspection as for Decision 2, we should truncate the tails of the distribution at asymmetric specification limits as shown in Fig. 11.9, where

$$\begin{aligned} \mathrm{LSL} &= \mu - n_1\sigma\,, \\ \mathrm{USL} &= \mu + n_2\sigma\,. \end{aligned}$$

Let $f_\mathrm{T}(y_\mathrm{T})$ be the probability density function for the truncated random variable Y_T; then we have

$$\begin{aligned} & f_\mathrm{T}(y_\mathrm{T}) = \frac{1}{q} f(y_\mathrm{T}) = \frac{1}{q\sigma\sqrt{2\pi}}\,\mathrm{e}^{-\frac{(y_\mathrm{T}-\mu)^2}{2\sigma^2}}\,, \\ & \text{where} \quad q = \Phi(n_1)+\Phi(n_2)-1\,, \\ & \text{and} \quad \mu - n_1\sigma \le y_\mathrm{T} \le \mu + n_2\sigma\,. \end{aligned}$$

Using the above information, we can prove that the expected quality loss for the truncation distribution is:

$$\begin{aligned} \mathcal{L}_\mathrm{T} = & \frac{1}{q}\left\{k_1\sigma\left[2(\mu-y_0)-n_1\sigma\right]\phi(n_1)\right. \\ & \left. + k_2\sigma\left[2(y_0-\mu)-n_2\sigma\right]\phi(n_2)\right\} \\ & + \frac{1}{q}\left\{\sigma(y_0-\mu)(k_1-k_2)\phi\left(\frac{y_0-\mu}{\sigma}\right)\right. \\ & \left. + (k_1-k_2)\left[\sigma^2+(y_0-\mu)^2\right]\Phi\left(\frac{y_0-\mu}{\sigma}\right)\right\} \\ & + \frac{1}{q}\left\{\left[\sigma^2+(y_0-\mu)^2\right]\right. \\ & \left. \times\left[k_1\Phi(n_1)+k_2\Phi(n_2)-k_1\right]\right\}. \end{aligned}$$

Then the expected quality loss per unit EQL is $q\mathcal{L}_T$, because the fraction of units shipped to customers is q. If k_1, k_2, y_0, ESC, and IC are given, we can minimize ETC_2 to find the optimal value of n_1, n_2, and the process mean value μ. The optimization model for Decision 2 is

$$\begin{aligned}\text{Min ETC}_2 &= q\mathcal{L}_T + (1-q)\text{SC} + \text{IC}\\ &= \Big\{ k_1\sigma[2(\mu - y_0) - n_1\sigma]\phi(n_1)\\ &\quad + k_2\sigma[2(y_0-\mu) - n_2\sigma]\phi(n_2)\\ &\quad + \sigma(y_0-\mu)(k_1-k_2)\phi\left(\frac{y_0-\mu}{\sigma}\right)\\ &\quad + (k_1-k_2)\left[\sigma^2 + (y_0-\mu)^2\right]\\ &\quad \times \Phi\left(\frac{y_0-\mu}{\sigma}\right)\\ &\quad + \left[\sigma^2 + (y_0-\mu)^2\right]\\ &\quad \times [k_1\Phi(n_1) + k_2\Phi(n_2) - k_1]\Big\}\\ &\quad + [2 - \Phi(n_1) - \Phi(n_2)]\,\text{SC} + \text{IC}\,.\end{aligned}$$

To choose from the alternative decisions, we should optimize the model for Decision 1 with zero inspection first and have the minimum expected total cost ETC_1^*. Then we optimize the model for Decision 2 with 100% inspection and have the optimal expected total cost ETC_2^*. If $ETC_1^* < ETC_2^*$, we should adjust the process mean to the optimal mean value given by the solutions and then ship all the output to the next or downstream customers without any inspection because the total cost to the system will be minimized in this way. Otherwise, we should take Decision 2, adjust the process mean, and do 100% inspection at the optimal specification limits given by the solutions of the optimization model.

For example, we need to make decisions in terms of the disposition of the output of a process that has the following parameters: the output of the process has a target value $y_0 = 10$; the quality loss function is asymmetrical about the target with $k_1 = 10$ and $k_2 = 5$, based on the input from the customer; the distribution of the process follows a normal distribution with $\sigma = 1.0$; the inspection cost per unit is IC = \$0.10, and the scrap cost per unit is SC = \$4.00. Should we do 100% inspection or zero inspection? If 100% inspection is to be done, what specification limits should be used?

First, we minimize the optimization model for Decision 1:

$$\begin{aligned}\text{Min ETC}_1 &= (k_1-k_2)\sigma(y_0-\mu)\phi\left(\frac{y_0-\mu}{\sigma}\right)\\ &\quad + \left[\sigma^2 + (y_0-\mu)^2\right]\\ &\quad \times \left[(k_1-k_2)\Phi\left(\frac{y_0-\mu}{\sigma}\right) + k_2\right]\\ &= 5(10-\mu)\phi\,(10-\mu)\\ &\quad + \left[1 + (10-\mu)^2\right]\\ &\quad \times [5\Phi\,(10-\mu) + 5]\end{aligned}$$

subject to $\mu \geq 0$.

Using Mathematica to solve the equation with the given set of parameters, we have the optimal solution $\mu^* = 10.28$, and $ETC_1^* = \$6.96$. Also, Genetic Algorithm by *Houck* et al. [11.15] gives us the same optimal solution.

Then, we optimize the model for Decision 2 given by

$$\begin{aligned}\text{Min ETC}_2 &= \Big\{(20\mu - 10n_1 - 200)\phi(n_1)\\ &\quad + (100 - 10\mu - 5n_2)\phi(n_2)\\ &\quad + (50 - 5\mu)\phi\,(10-\mu)\\ &\quad + \left[5 + 5(10-\mu)^2\right]\Phi\,(10-\mu)\\ &\quad + \left[1 + (10-\mu)^2\right]\\ &\quad \times [10\Phi(n_1) + 5\Phi(n_2) - 10]\Big\}\\ &\quad + 4\,[2 - \Phi(n_1) - \Phi(n_2)] + 0.1\end{aligned}$$

subject to $n_1 \geq 0,\ n_2 \geq 0$.

This can be minimized using Genetic Algorithm provided by *Houck* et al. [11.15], which gives us $n_1^* = 0.72$, $n_2^* = 0.82$, $\mu^* = 10.08$, and $TC^* = \$2.57 < \$6.96$.

Since $ETC_1^* > ETC_2^*$, we should adjust the process mean to 10.08 given by the optimal solution from Decision 2 and do 100% inspection with respect to LSL = 9.36 and USL = 10.90 to screen the nonconforming units. In this way, the expected total cost to the whole system will result in a reduction of \$4.39, or 63% decrease in ETC. This example presents a way to determine the optimal process mean value and specification limits by minimizing the total cost to both producer and consumer.

11.3 Process Optimization

In the previous section, it is assumed that it is difficult to improve the process because of the constraint of the current technology, cost, or capability. To improve the performance of the output, we screen or inspect the product before shipping to the customer by setting up optimal specification limits on the distribution of the output. Thus the focus is on inspection of the product. To further optimize the performance of the system, it is supposed that the process can be improved during the design phase, which is also called offline quality engineering. Then the process should be designed and optimized with any effort to meet the requirements of customers economically. During offline quality engineering, three design phases need to be taken [11.7]:

- ***System design***: The process is selected from knowledge of the pertinent technology. After system design, it is often the case that the exact functional relationship between the output variables and input variables cannot be expressed analytically. One needs to explore the functional relationship of the system empirically. Design of experiments is an important tool to derive this system transfer function. Orthogonal polynomial expansion also provides an effective means of evaluating the influences of input variables on the output response.
- ***Parameter design***: The optimal settings of input variables are determined to optimize the output variable by reducing the influence of noise factors. This phase of design makes effective use of experimental design and response surface methods.
- ***Tolerance design***: The tolerances or variances of the input variables are set to minimize the variance of output response by directly removing the variation causes. It is usually true that a narrower tolerance corresponds to higher cost. Thus cost and loss due to variability should be carefully evaluated to determine the variances of input variables. Experimental design and response surface methods can be used in this phase.

In the following sections, the statistical methods involved in the three design phases are reviewed, including experimental design method, orthogonal polynomial expansion, and response surface method. Since the ultimate goal is to minimize the total cost to both producers and consumers, or the whole system, some integrated optimization models are developed from the system point of view.

11.3.1 Design of Experiments

Introduction to Design of Experiments

Experiments are typically operations on natural entities and processes to discover their structure, functioning, or relationships. They are an important part of the scientific method, which entails observation, hypothesis, and sequential experimentation. In fact, experimental design methods provide us the tools to test the hypothesis, and thus to learn how systems or processes work.

In general, experiments are designed to study the performance of processes or systems. The process or system model can be illustrated by Fig. 11.2 as given in the introduction of this chapter. The process consists of many input variables and one or multiple output variables. The input variables include both controllable factors and uncontrollable or noise factors.

Experimental design methods have broad applications in many disciplines such as agriculture, biological and physical sciences, and design and analysis of engineering systems. They can be used to improve the performance of existing processes or systems and also to develop new ones. The applications of experimental design techniques can be found in:

- Improving process yields
- Reducing variability including both bias from target value and variance
- Evaluating the raw material or component alternatives
- Selecting of component-level settings to make the output variables robust
- Reducing the total cost to the organization and/or the customer

Procedures of Experimental Design

To use statistical methods in designing and analyzing an experiment, it is necessary for experimenters to have a clear outline of procedures as given below.

Problem Statement or Definition. A clear statement of the problem contributes substantially to better understanding the background, scope, and objective of the problem. It is usually helpful to list the specific problems that are to be solved by the experiment. Also, the physical, technological, and economic constraints should be stated to define the problem.

Selection of Response Variable. After the statement of the problem, the response variable y should be selected. Usually, the response variable is a key performance measurement of the process, or the critical-to-quality (CTQ) characteristic. It is important to have precise measures of the response variable. If at all possible, it should be a quantitative (variable) quality characteristic, which would make data analysis easier and meaningful.

Choice of Factors, Levels, and Ranges. Cause and effect diagrams should be developed by a team or panels of experts in the area. The team should represent all points of view and should also include people necessary for implementation. A brainstorming approach can be used to develop theories for the construction of cause and effect diagrams.

From the cause and effect diagrams a list of factors that affect the response variables is developed, including both qualitative and quantitative variables. Then the factors are decomposed into control factors and noise factors. Control factors are factors that are economical to control. Noise factors are uncontrollable or uneconomical to control. Three types of noise factors are outer noise, inner noise, and production noise.

The list of factors is generally very large, and the group may have to prioritize the list. The number of factors to include in the study depends on the priorities, difficulty of experimentation, and budget. The final list should include as many control factors as possible and some noise factors that tend to give high or low values of the response variable.

Once the factors have been selected, the experimenter must choose the number of levels and the range for each factor. It also depends on resource and cost considerations. Usually, factors that are expected to have a linear effect can be assigned two levels, while factors that may have a nonlinear effect should have three or more levels. The range over which the factors are varied should also be chosen carefully.

Selection of Experimental Design. The selection of experimental design depends on the number of factors, the number of levels for each factor, and the number of replicates that provides the data to estimate the experimental error variance. Also, the determination of randomization restrictions is involved, such as blocking or not. Randomization justifies the statistical inference methods of estimation and tests of hypotheses. In selecting the design, it is important to keep the experimental objectives in mind. Several books review and discuss the types of experimental designs and how to choose an appropriate experimental design for a wide variety of problems [11.16–18].

Conduction of the Experiment. Before performing the experiment, it is vital to make plans for special training if required, design data sheets, and schedule for experimentation etc. In the case of product design experimentation, sometimes the data can be collected through the use of simulation programs rather than experiments with actual hardware. Then the computer simulation models need to be developed before conducting the experiment.

When running the experiment in the laboratory or a full-scale environment, the experimenter should monitor the process on the right track, collect all the raw data, and record unexpected events.

Analysis and Interpretation of the Data. Statistical methods are involved in data analysis and interpretation to obtain objective conclusions from the experiment. There are many software packages designed to assist in data analysis, such as SAS, S-Plus, etc. The statistical data analysis can provide us with the following information:

- Which factors and interactions have significant influences on the response variable?
- What are the rankings of relative importance of main effects and interactions?
- What are the optimal factor level settings so that the response is near the target value? (parameter design)
- What are the optimal factor level settings so that the effects of the noise factors are minimized? (robust design)
- What are the best factor level settings so that the variability of the response is reduced?
- What is the functional relationship between the controllable factors and response, or what is the empirical mathematical model relating the output to the input factors?

Statistical methods lend objectivity to the decision-making process and attach a level of confidence to a statement. Usually, statistical techniques will lead to solid conclusions with engineering knowledge and common sense.

Conclusions and Recommendations. After data analysis, the experimenter should draw some conclusions and recommend an action plan. Usually, a confirmation experiment is run to verify the reproducibility of the optimum recommendation. If the result is not confirmed

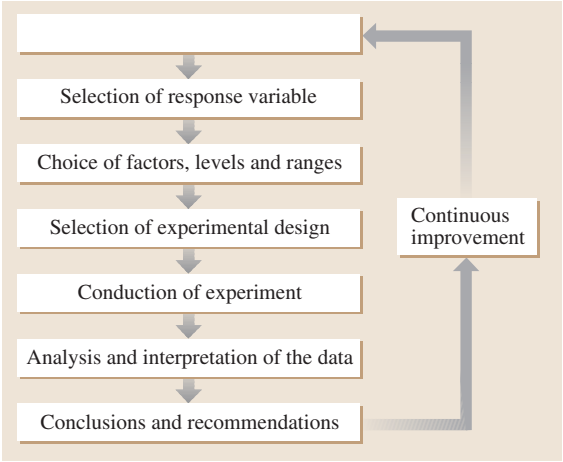


Fig. 11.10 Iterative procedures of experimental design

or is unsatisfactory, additional experimentation may be required.

Based on the results of the confirmation experiment and the previous analysis, the experimenter can develop sound conclusions and recommendations.

Continuous Improvement. The entire process is actually a learning process, where hypotheses about a problem are tentatively formulated, experiments are conducted to investigate these hypotheses, and new hypotheses are then formulated based on the experimental results. By continuous improvement, this iterative process moves us closer to the "truth" as we learn more about the system at each stage (Fig. 11.10). Statistical methods enter this process at two points: (1) selection of experimental design and (2) analysis and interpretation of the data [11.16].

11.3.2 Orthogonal Polynomials

Most research in engineering is concerned with the derivation of the unknown functional relationship between input variables and output response. In many cases, the model is often easily and elegantly constructed as a series of orthogonal polynomials [11.19–21]. Compared with other orthogonal functions, the orthogonal polynomials are particularly convenient for at least two reasons. First, polynomials are easier to work with than irrational or transcendental functions; second, the terms in orthogonal polynomials are statistically independent, which facilitates both their generation and processing. One of the other advantages of orthogonal polynomials is that users can simply develop their own system of functions in accordance with the particular problem. More often, a problem can be transformed to one of the standard families of polynomials, for which all significant relations have already been worked out.

Orthogonal polynomials can be used whether the values of controllable factors Xs are equally or unequally spaced [11.22]. However, the computation is relatively easy when the values of factor levels are in equal steps. For a system with only one equal-step input variable X, the general orthogonal polynomial model of the functional relationship between response variable Y and X is given as

$$y = \mu + \alpha_1 P_1(x) + \alpha_2 P_2(x) + \alpha_3 P_3(x) + \cdots + \alpha_n P_n(x) + \varepsilon \,, \tag{11.2}$$

where x is the value of factor level, y is the measured response [11.17], μ is the grand mean of all responses, and $P_k(x)$ is the kth-order orthogonal polynomial of factor X. The transformations for the powers of x into orthogonal polynomials $P_k(x)$ up to the cubic degree are given below:

$$P_1(x) = \lambda_1 \left(\frac{x-\bar{x}}{d}\right) ,$$

$$P_2(x) = \lambda_2 \left[\left(\frac{x-\bar{x}}{d}\right)^2 - \left(\frac{t^2-1}{12}\right)\right] ,$$

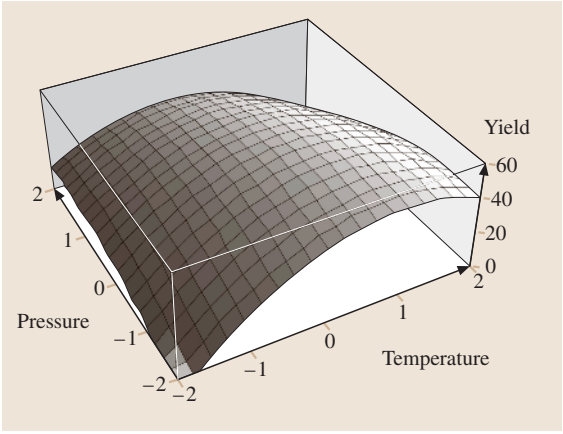
$$P_3(x) = \lambda_3 \left[\left(\frac{x-\bar{x}}{d}\right)^3 - \left(\frac{x-\bar{x}}{d}\right)\left(\frac{3t^2-7}{20}\right)\right] , \tag{11.3}$$

where $\bar{x}$ is the average value of factor levels, t is the number of levels of the factor, d is the distance between factor levels, and the constant λ_k makes $P_k(x)$ an integral value for each x.

Since t, d, $\bar{x}$, and x are known, $P_k(x)$ can be calculated for each x. For example, a four-level factor X ($t = 4$) can fit a third-degree equation in x. The orthogonal polynomials can be tabulated based on the calculation of (11.3) as below:

	$P_1(x)$	$P_2(x)$	$P_3(x)$
x_1	−3	1	−1
x_2	−1	−1	3
x_3	1	−1	−3
x_4	3	1	1

The values of the orthogonal polynomials $P_k(x)$ have been tabulated up to $t = 104$ [11.21].



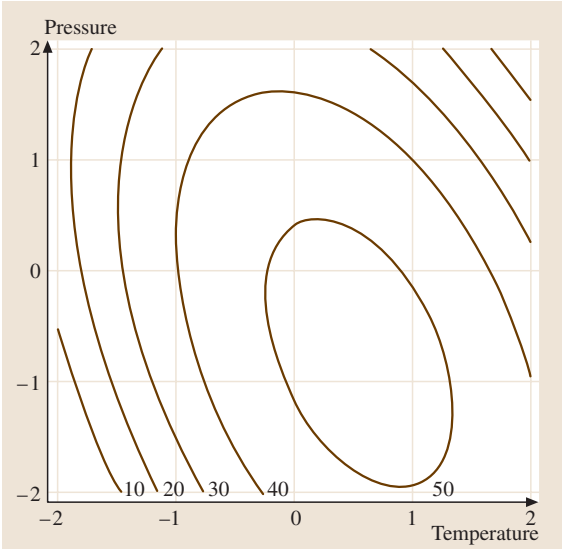


Fig. 11.11a,b Response surface **(a)** and contour plot **(b)** for a chemical process

Given the response y_i for the ith level of X, x_i, $i = 1, 2, \ldots, t$, the estimates of the α_k coefficients for the orthogonal polynomial (11.2) are calculated as

$$\alpha_k = \sum_{i=1}^{t} y_i P_k(x_i) \Big/ \sum_{i=1}^{t} P_k(x_i)^2$$

for $k = 1, 2, \ldots, n$. The estimated orthogonal polynomial equation is found by substituting the estimates of $\mu, \alpha_1, \alpha_2, \cdots, \alpha_n$ into (11.2).

It is desirable to find the degree of polynomials that adequately represents the functional relationship between the response variable and the input variables. One strategy to determine the polynomial equation is to test the significance of the terms in the sequence: linear, quadratic, cubic, and so forth. Beginning with the simplest polynomial, a more complex polynomial is constructed as the data require for adequate description. The sequence of hypotheses is $H_0 : \alpha_1 = 0$, $H_0 : \alpha_2 = 0$, $H_0 : \alpha_3 = 0$, and so forth. These hypotheses about the orthogonal polynomials are each tested with the F test ($F = \mathrm{MSC}/\mathrm{MSE}$) for the respective polynomial. The sum of square for each polynomial needs to be calculated for the F test, which is

$$SS_{P_k} = \left(\sum_{i=1}^{t} y_i P_k(x_i) \right)^2 \Big/ \sum_{i=1}^{t} P_k(x_i)^2$$

for $k = 1, 2, \ldots, n$. The system function relationship can be developed by including the statistically significant terms in the orthogonal polynomial model.

For the multiple equal-step input variables $X_1, X_2, \ldots, X_n$, the orthogonal polynomial equation is found in a similar manner as for the single input variable. *Kuel* [11.17] gives an example of water uptake by barley plants to illustrate procedures to formulate the functional relationship between the amount of water uptake and two controllable factors: salinity of medium and age of plant.

11.3.3 Response Surface Methodology

Response surface methodology (RSM) is a specialized experimental design technique for developing, improving, and optimizing products and processes. The method can be used in the analysis and improvement phases of the (D)MAIC(T) process. As a collection of statistical and mathematical methods, RSM consists of an experimental strategy for exploring the settings of input variables, empirical statistical modeling to develop an appropriate approximating relationship between the response and the input variables, and optimization methods for finding the levels or values of the input variables that produce desirable response values.

Figure 11.11 illustrates the graphical plot of response surface and the corresponding contour plot for a chemical process, which shows the relationship between the response variable yield and the two process variables: temperature and pressure. Thus, when the response surface is developed by the design of experiments and constructed graphically, optimization of the process becomes easy using the response surface.

The process model given in Fig. 11.2 is also very useful for RSM. Through the response surface methodology, it is desirable to make the process box "transparent" by obtaining the functional relationship between the output response and the input factors. In fact, successful use of RSM is critically dependent upon the development of a suitable response function. Usually, either a first-order or second-order model is appropriate in a relatively small region of the variable space.

In general, a first-order response model can be written as

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_n X_n + \varepsilon \, .$$

For a system with nonlinear behavior, a second-order response model is used as given below:

$$Y = b_0 + \sum_i b_i X_i + \sum_i b_{ii} X_i^2 + \sum_i \sum_j d_{ij} X_i X_j + \varepsilon \, .$$

The method of least squares estimation is used to estimate the coefficients in the above polynomials. The second-order model is widely used in response surface methodology.

As an extended branch of experimental design, RSM has important applications in the design, development, and formulation of new products, as well as in the improvement of existing product designs. The applications of RSM can be found in many industrial settings where several variables influence the desired outcome (e.g., minimum fraction defective or maximum yield), including the semiconductor, electronic, automotive, chemical, and pharmaceutical industries.

Sequential Procedures of RSM

The applications of RSM are sequential in nature [11.23]. That is, at first we perform a screening experiment to reduce the list of candidate variables to a relatively few, so that subsequent experiments will be more efficient and require few tests. Once the important independent variables are identified, the next objective is to determine if the current levels or settings of the independent variables result in a value of the response that is near the optimum. If they are not consistent with optimum performance, a new set of adjustments to input variables should be determined to move the process toward the optimum. When the process is near the optimum, a model is needed to accurately approximate the true response function within a relatively small region around the optimum. Then, the model can be analyzed to identify the optimum conditions for the process. We can list the sequential procedures as follows [11.24]:

Step 0: *Screening experiment.* Usually the list of input variables is rather long, and it is desirable to start with a screening experiment to identify the subset of important variables. After the screening experiment, the subsequent experiments will be more efficient and require fewer runs or tests.

Step 1: *Determine if the optimal solution is located inside the current experimental region.* Once the important variables are identified through screening experiments, the experimenter's objective is to determine if the current settings of the input variables result in a value of response that is near optimum. If the current settings are not consistent with optimum performance, then go to step 2; otherwise, go to step 3.

Step 2: *Search the region that contains the optimal solution.* The experimenter must determine a set of adjustments to the process variables that will move the process toward the optimum. This phase of response surface methodology makes considerable use of the first-order model with two-level factorial experiment, and an optimization technique called the method of steepest ascent. Once the region containing the optimum solution is determined, go to step 3.

Step 3: *Establish an empirical model to approximate the true response function within a relatively small region around the optimum.* The experimenter should design and conduct a response surface experiment and then collect the experimental data to fit an empirical model. Because the true response surface usually exhibits curvature near the optimum, a nonlinear empirical model (often a second-order polynomial model) will be developed.

Step 4: *Identify the optimum solution for the process.* Optimization methods will be used to determine the optimum conditions. The techniques for the analysis of the second-order model are presented by *Myers* [11.23].

The sequential nature of response surface methodology allows the experimenter to learn about the process or system as the investigation proceeds. The investigation procedures involve several important topics/methods, including two-level factorial designs, method of steepest ascent, building an empirical model, analysis of second-order response surface, and response surface experimental designs, etc. For more detailed information, please refer to *Myers* [11.23] and *Yang* and *El-Haik* [11.24].

11.3.4 Integrated Optimization Models

The ultimate objective of Six Sigma strategy is to minimize the total cost to both producer and consumer, or the whole system. The cost to the consumer is related to the expected quality loss of the output variable, and it is caused by the deviation from the target value. The cost to the producer is associated with changing probability distributions of input variables. If the system transfer function and the variance transmission equation are available, and the cost functions for different grades of input factors are given, the general optimization model to reflect the optimization strategy is given in Fig. 11.12.

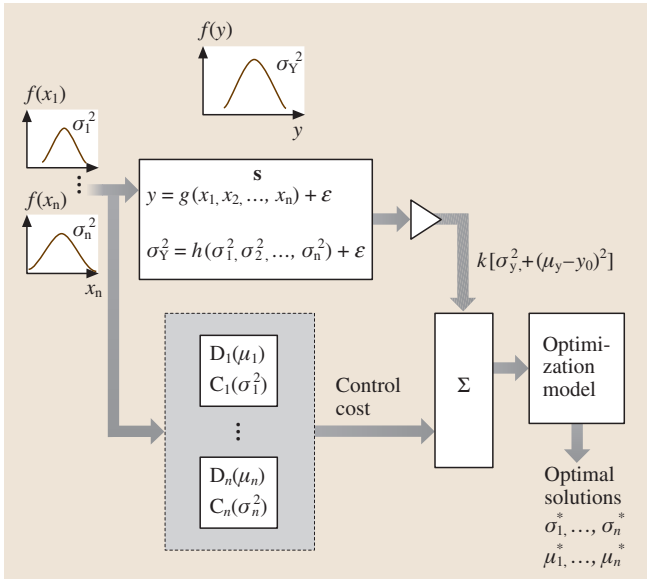


Fig. 11.12 General optimization model for system

General Optimization Problem

We usually consider the first two moments of the probability distributions of input variables, and then the optimization models will focus on the mean and variance values. Therefore, the expected quality loss to the consumer consists of two parts: the bias of the process and the variance of the process. The strategy to reduce bias is to find adjustment factors that do not affect variance and thus are used to bring the mean closer to the target value. Design of experiments can be used to find these adjustment factors. It will incur certain costs to the producer. To reduce the variance of Y, the designer should reduce the variances of the input variables, which will also increase costs. The problem is to balance the reduced expected quality loss with the increased cost for the reduction of the bias and variances of the input variables. Typically, the variance control cost for the ith input variable X_i is denoted by $C_i(\sigma_i^2)$, and the mean control cost for the ith input variable X_i is denoted by $D_i(\mu_i)$. By focusing on the first two moments of the probability distributions of $X_1, X_2, \ldots, X_n$, the general optimization model is formulated as

$$\begin{aligned}\text{Minimize TC} = &\sum_{i=1}^{n} C_i\left(\sigma_i^2\right) + \sum_{i=1}^{n} D_i(\mu_i) \\ &+ k\left[\sigma_Y^2 + (\mu_Y - y_0)^2\right] , \\ \text{subject to } &\mu_Y \approx m(\mu_1, \mu_2, \cdots, \mu_n) , \\ &\sigma_Y^2 \approx h\left(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2\right) . \end{aligned} \tag{11.4}$$

In this objective function, the first two terms,

$$\sum_{i=1}^{n} C_i\left(\sigma_i^2\right) \text{ and } \sum_{i=1}^{n} D_i(\mu_i) ,$$

are the control costs on the variances and means of input variables, or the cost to the producer; the last term, $k\left[\sigma_Y^2 + (\mu_Y - y_0)^2\right]$, is the expected quality loss to the customer, where k is a constant in the quality loss function. The first constraint, $\mu_Y \approx m(\mu_1, \mu_2, \cdots, \mu_n)$, is the model for the mean of the system, which can be obtained through the system transfer function. The second constraint, $\sigma_Y^2 \approx h(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2)$, is the variance transmission equation. A future research problem is to solve this optimization problem in such a way as to consider together both the mean and the variance.

Tolerance Design Problem

If we assume that the bias reduction has been accomplished, the general optimization problem given by (11.4) can be simplified as a tolerance design problem, which is given below:

$$\begin{aligned}\text{Minimize TC} = &\sum_{i=1}^{n} C_i\left(\sigma_i^2\right) + k\sigma_Y^2 \\ \text{subject to } &\sigma_Y^2 \approx h\left(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2\right) . \end{aligned} \tag{11.5}$$

The objective of the tolerance design is to determine the tolerances (which are related to variances) of the input variables to minimize the total cost, which consists of the expected quality loss due to variation $k\sigma_Y^2$ and the control cost on the tolerances of the input variables

$$\sum_{i=1}^{n} C_i\left(\sigma_i^2\right) .$$

Typically, $C_i(\sigma_i^2)$ is a nonincreasing function of each σ_i^2.

For this tolerance design problem, a RLC circuit example is given by *Chen* [11.25] to minimize the total cost to both the manufacturer and the consumer. Taguchi's method is used to construct the variance transmission equation as the constraint in Chen's example. *Bare* et al. [11.26] propose another optimization model to minimize the total variance control cost by finding the optimum standard deviations of input variables. Taylor's series expansion is used to develop the variance transmission equation in their model.

Case Study: Wheatstone Bridge Circuit Design

We use the Wheatstone bridge circuit design problem [11.7] as a case study to illustrate models described above [11.27]. The system transfer function is known for this example, and thus we will illustrate the development of variance transmission equation and optimization design models.

The Wheatstone bridge in Fig. 11.13 is used to determine an unknown resistance Y by adjusting a known resistance so that the measured current is zero. The resistor B is adjusted until the current X registered by the galvanometer is zero, at which point the resistance value B is read and Y is calculated from the formula $Y = BD/C$. Due to the measurement error, the current is not exactly zero, and it is assumed to be a positive or negative value of about 0.2 mA. In this case the resistance is given by the following system transfer function:

$$Y = \frac{BD}{C} - \frac{X}{C^2 E}[A(C+D)+D(B+C)] \times [B(C+D)+F(B+C)] \ .$$

The noise factors in the problem are variability of the bridge components, resistors A, C, D, F, and input voltage E. This is the case where control factors and noise factors are related to the same variables. Another noise factor is the error in reading the galvanometer X. Assuming that when the galvanometer is read as zero, there may actually be a current about 0.2 mA. Taguchi did the parameter design using L_{36} orthogonal arrays for the design of the experiment. When the parameter design cannot sufficiently reduce the effect of internal and external noises, it becomes necessary to develop the variance transmission equation and then control the variation of the major noise factors by reducing their tolerances, even though this increases the cost.

Let the nominal values or mean of control factors be the second level and the deviations due to the noise factors be the first and third level. The three levels of noise factors for the optimum combination based on parameter design are given in Table 11.1.

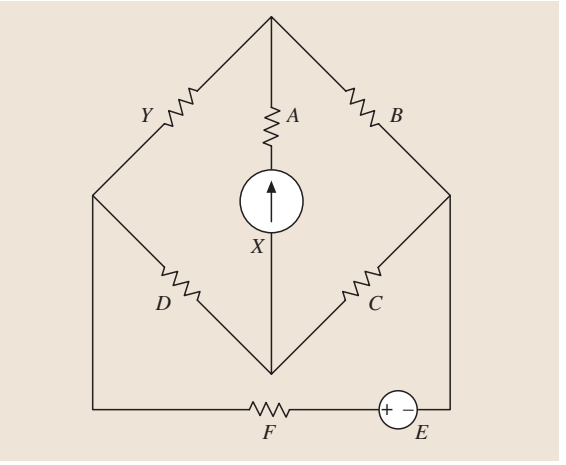


Fig. 11.13 Wheatstone bridge and parameter symbols

We use three methods to develop the variance transmission equation: Taylor series approximation, response surface method, and experimental design method. The results for various approaches are given in Table 11.2. RSM (L_{36}) and DOE (L_{36}) have the same L_{36} orthogonal array design layout for comparison purposes. Improved RSM and improved DOE use the complete design with $N = 3^7 = 2187$ design points for the unequal-mass three-level noise factors. For comparison purposes, we also perform the complete design with 2187 data points for the equal-mass three-level noise factors, which are denoted as RSM (2187) and DOE (2187) in Table 11.2. Without considering the different design layouts, it seems that the improved method gives better approximation of variance. We can see that the improved DOE's VTE does not differ much from the original one in its ability to approximate the variance of the response. Because the improved DOE method requires the complete evaluation at all combinations of levels, it is costly in terms of time and resources. If

Table 11.1 Noise factor levels for optimum combination

Factor	Level 1	Level 2	Level 3
$A(\Omega)$	19.94	20	20.06
$B(\Omega)$	9.97	10	10.03
$C(\Omega)$	49.85	50	50.15
$D(\Omega)$	9.97	10	10.03
$E(\mathrm{V})$	28.5	30	31.5
$F(\Omega)$	1.994	2	2.006
$X(\mathrm{A})$	−0.0002	0	0.0002

Table 11.2 Comparison of results from different methods

Methods	VTE	σ_Y^2
Linear Taylor	$\sigma_Y^2 = 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,00\sigma_D^2 + 276.002\,84\sigma_X^2 + O(\sigma^3)$	$7.939\,01 \times 10^{-5}$
Nonlinear Taylor	$\sigma_Y^2 = 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,00\sigma_D^2 + 276.002\,84\sigma_X^2 + 3.84 \times 10^{-6}\sigma_C^4 + O(\sigma^5)$	$7.939\,10 \times 10^{-5}$
RSM (L_{36})	$\sigma_Y^2 = 0.040\,04\sigma_B^2 + 0.001\,62\sigma_C^2 + 0.040\,36\sigma_D^2 + 300.373\,96\sigma_X^2 + 1.42 \times 10^{-8}$	$8.045\,28 \times 10^{-5}$
RSM (2187)	$\sigma_Y^2 = 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,00\sigma_D^2 + 299.598\,75\sigma_X^2 + 1.00 \times 10^{-8}$	$8.000\,03 \times 10^{-5}$
IPV RSM	$\sigma_Y^2 = 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,00\sigma_D^2 + 299.601\,30\sigma_X^2 + 1.43 \times 10^{-8}$	$8.000\,51 \times 10^{-5}$
DOE (L_{36})	$\sigma_Y^2 = 0.041\,18\sigma_B^2 + 0.001\,66\sigma_C^2 + 0.041\,50\sigma_D^2 + 308.939\,35\sigma_X^2 + 1.42 \times 10^{-8}$	$8.274\,94 \times 10^{-5}$
DOE (2187)	$\sigma_Y^2 = 0.040\,02\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,02\sigma_D^2 + 299.735\,65\sigma_X^2 + 1.00 \times 10^{-8}$	$8.004\,08 \times 10^{-5}$
IPV DOE	$\sigma_Y^2 = 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,00\sigma_D^2 + 299.768\,00\sigma_X^2 + 1.43 \times 10^{-8}$	$8.000\,95 \times 10^{-5}$
Monte Carlo	1 000 000 observations	$7.998\,60 \times 10^{-5}$

Note: The calculation of σ_Y^2 is for $\sigma_B = 0.024\,49$, $\sigma_C = 0.122\,47$, $\sigma_D = 0.024\,49$, $\sigma_X = 0.000\,16$; RSM (2187) is the response surface method applied on the same data set as Taguchi's VTE (2187); improved (IPV) RSM is the response surface method applied on the same data set as the improved (IPV) Taguchi VTE

the high cost of the complete design is a concern, the original DOE's equal-mass three-level method using orthogonal array is preferred. If the complete evaluation can be accomplished by simulation without much difficulty, the improved DOE method should be applied to ensure high accuracy. Thus, the variance transmission equation for this Wheatstone bridge circuit is determined as

$$\sigma_Y^2 = 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 + 0.040\,00\sigma_D^2 + 299.768\,00\sigma_X^2 + 1.43 \times 10^{-8}\,.$$

For such a problem, we can easily develop the mean model and use it with the above VTE to develop the general optimization model. It is well understood that the tolerances or variances on resistors, voltage, and current impact the cost of the design, i. e., tighter tolerances result in higher cost. Thus we can develop the variance control cost functions $C_i(\sigma_i^2)$ for each component. Similarly, the mean control cost functions $D_i(\mu_i)$ for any problem can be developed. For this problem, if the cost associated with changing the mean values is relatively small or insignificant, then we can just focus on the tolerance design problem given by (11.5), which is

$$\begin{aligned}\text{Minimize TC} &= C_B(\sigma_B^2) + C_C(\sigma_C^2) + C_D(\sigma_D^2) \\ &\quad + C_X(\sigma_X^2) + k\sigma_Y^2\,, \\ \text{subject to } \sigma_Y^2 &= 0.040\,00\sigma_B^2 + 0.001\,60\sigma_C^2 \\ &\quad + 0.040\,00\sigma_D^2 + 299.768\,00\sigma_X^2 \\ &\quad + 1.43 \times 10^{-8}\,.\end{aligned}$$

Based on the complexity of the cost functions $C_i(\sigma_i^2)$ and $D_i(\mu_i)$ and the constraint, such optimization problems can be solved by many optimization methods including software available for global search algorithms such as genetic algorithm optimization toolbox (GAOT) for Matlab 5 (http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/).

11.4 Summary

In this chapter, we first introduce the Six Sigma quality and design for Six Sigma process. By focusing on the analysis and improvement phases of the (D)MAIC(T) process, we discuss the statistical and optimization strategies for product and process optimization, respectively. Specifically, for product optimization, we review the quality loss function and various optimization models for specification limits development. For process optimization, we discuss design of experiments, orthogonal polynomials, response surface methodology, and integrated optimization models. Those statistical methods play very important roles in the activities for process and product improvement.

References

11.1 Motorola University: *Home of Six Sigma methodology and practice* ((Online) Motorola Inc. Available from: https://mu.motorola.com/, Accessed 27 May 2004)

11.2 General Electric Company: *What is Six Sigma: The Roadmap to Customer Impact* ((Online) General Electric Company. Available from: http://www.ge.com/sixsigma/, Accessed 27 May 2004)

11.3 F. W. Breyfogle: *Implementing Six Sigma: Smarter Solutions Using Statistical methods, 2nd edn.* (Wiley, New York 2003)

11.4 W. E. Deming: *Quality, Productivity, and Competitive Position* (MIT, Center for Advanced Engineering Study, Cambridge 1982)

11.5 J. Orsini: Simple rule to reduce total cost of inspection and correction of product in state of chaos, Ph.D. Dissertation, Graduate School of Business Administration, New York University (1982)

11.6 E. P. Papadakis: The Deming inspection criterion for choosing zero or 100 percent inspection, J. Qual. Technol. **17**, 121–127 (1985)

11.7 G. Taguchi: *Introduction to Quality Engineering* (Asia Productivity Organization, Tokyo 1986)

11.8 G. Taguchi: *System of Experimental Design, Volume I and II, Quality Resources* (American Supplier Institute, Deaborn, MI 1987)

11.9 K. C. Kapur: *Quality Loss Function and Inspection*, Proc. TMI Conf. Innovation in Quality (Engineering Society of Detroit, Detroit, 1987)

11.10 K. C. Kapur, D. J. Wang: *Economic Design of Specifications Based on Taguchi's Concept of Quality Loss Function*, Proc. Am. Soc. Mech. Eng. (ASME, Boston, 1987)

11.11 K. C. Kapur, B. Cho: Economic design and development of specifications, Qual. Eng. **6(3)**, 401–417 (1994)

11.12 K. C. Kapur, B. Cho: Economic design of the specification region for multiple quality characteristics, IIE Trans. **28**, 237–248 (1996)

11.13 K. C. Kapur: An approach for development of specifications for quality improvement, Qual. Eng. **1(1)**, 63–78

11.14 Q. Feng, K. C. Kapur: Economic development of specifications for 100% inspection based on asymmetric quality loss function, IIE Trans. Qual. Reliab. Eng. (2003) in press

11.15 C. R. Houck, J. A. Joines, M. G. Kay: A Genetic Algorithm for Function Optimization: A Matlab Implementation, NCSU-IE Technical Report, 95-09, 1995

11.16 C. R. Hicks, K. V. Turner: *Fundamental Concepts in the Design of Experiments, 5th edn.* (Oxford University Press, New York 1999)

11.17 R. O. Kuehl: *Statistical Principles of Research Design and Analysis* (Duxbury Press, Belmont, CA 1994)

11.18 D. C. Montgomery: *Design and Analysis of Experiments, 5th edn.* (Wiley, New York 2001)

11.19 F. S. Acton: *Analysis of Straight-Line Data* (Wiley, New York 1959)

11.20 P. Beckmann: *Orthogonal Polynomials for Engineers and Physicists* (Golem Press, Boulder, CO 1973)

11.21 F. A. Graybill: *An Introduction to Linear Statistical Models* (McGraw-Hill, New York 1961)

11.22 A. Grandage: Orthogonal coefficients for unequal intervals, query 130, Biometrics **14**, 287–289 (1958)

11.23 R. H. Myers, D. C. Montgomery: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments* (Wiley, New York 2002)

11.24 K. Yang, B. El-Haik: *Design for Six Sigma: A Roadmap for Product Development* (McGraw-Hill, New York 2003)

11.25 G. Chen: Product and process design optimization by quality engineering, Ph.D. Dissertation, Wayne State University, Detroit (1990)

11.26 J. M. Bare, K. C. Kapur, Z. B. Zabinsky: Optimization methods for tolerance design using a first-order approximation for system variance, Eng. Design Autom. **2**, 203–214 (1996)

11.27 K. C. Kapur, Q. Feng: Integrated optimization models and strategies for the improvement of the six sigma process, Int. J. Six Sigma Comp. Adv. **1**(2) (2005)

12. Robust Optimization in Quality Engineering

Quality engineers often face the job of identifying process or product design parameters that optimize performance response. The first step is to construct a model, using historical or experimental data, that relates the design parameters to the response measures. The next step is to identify the best design parameters based on the model. Clearly, the model itself is only an approximation of the true relationship between the design parameters and the responses. The advances in optimization theory and computer technology have enabled quality engineers to obtain a good solution more efficiently by taking into account the inherent uncertainty in these empirically based models.

Two widely used techniques for parameter optimization, described with examples in this chapter, are the response surface methodology (RSM) and Taguchi loss function. In both methods, the response model is assumed to be fully correct at each step. In this chapter we show how to enhance both methods by using robust optimization tools that acknowledge the uncertainty in the models to find even better solutions. We develop a family of models from the confidence region of the model parameters and show how to use sophistical optimization techniques to find better design parameters over the entire family of approximate models.

Section 12.1 of the chapter gives an introduction to the design parameter selection problem and motivates the need for robust optimization. Section 12.2 presents the robust optimization approach to address the problem of optimizing empirically based response functions by developing a family of models from the confidence region of the model parameters. In Sect. 12.2 robust optimization is compared to traditional optimization approaches where the empirical model is assumed to be true and the optimization is conducted without considering the uncertainty in the parameter estimates. Simulation is used to make the comparison in the context of response surface methodology, a widely used method to optimize products and processes that is briefly described in the section. Section 12.3 introduces a refined technique, called weighted robust optimization, where more-likely points in the confidence region of the empirically determined parameters are given heavier weight than less-likely points. We show that this method provides even more effective solutions compared to robust optimization without weights. Section 12.4 discusses Taguchi's loss function and how to leverage robust optimization methods to obtain better solutions when the loss function is estimated from empirical experimental data.

One of the central themes in quality engineering is the identification of optimal values for the design parameters to make a process or product function in the best possible way to maximize its performance. The advances in optimization theory and computing technology in the last half century have greatly stimulated the progress in quality improvement—optimization methodology has provided a systematic framework to guide today's quality engineers to identify optimal levels in design parameters efficiently, while the same task would have taken many iterations of experiments for engineers one generation ago without the aid of modern optimization techniques.

Many quality engineering problems arising in today's complex manufacturing processes can be reduced to some optimization problem. For example, in process control problems, we are interested in selecting a best possible set of values for process settings to maximize the output of the final products that satisfy the specifications in the shortest amount of time. In the context of product design problems, the purpose is to choose an optimal mix of design parameters to maximize the performance measures of the new products.

The iteration process in applying optimization techniques to solve quality improvement problems includes the following steps:

1. Convert the quality requirements and specifications to an optimization model; (12.1)
2. Solve the optimization problems and identify the optimal values for the decision variables, i. e., the process settings or design parameters;
3. Apply the optimal solution identified in step 2 to the actual process control or product design environment, validate the effectiveness of the optimal solution and revise the optimization model if necessary.

There exists a large volume of literature advocating the use of optimization techniques to improve process and product quality; see *Box* et al. [12.1], *Box* and *Draper* [12.2], *Myers* and *Montgomery* [12.3], *Khuri* and *Cornell* [12.4], among many others.

The most critical step in the above procedure is to construct the optimization model using the historical or experimental data collected in the process control or product design stage. Usually we tend to regard a model constructed on empirical data as a true physical law. Thus we assume that the model accurately describes the underlying process or product and that the optimal solution to the model is better than any other choice.

However there is much uncertainty involved in the model construction process. First, the most common uncertainty comes from the measurement error and noise effect. The devices used to capture the readings are more or less subject to measurement errors. Noise factors, such as environmental conditions and material properties, will sometimes severely distort the values of the true performance measure. Second, the failure to identify and record every possible main factor that contributes to the final performance measure will certainly degrade the quality of the model since it cannot incorporate all of the major predictors. Finally the model selection process adds another layer of uncertainty in the final model we will reach. There are numerous forms of models we can choose from. For example, should we develop a linear model or a nonlinear one? If it is a nonlinear model, should we try a higher-order polynomial function or a logistic function, or something else?

The uncertainty in the model construction process poses huge challenges to the statistical sciences, which have provided numerous methods to identify effective models to represent the true relationship between the design parameters and process/product performance as closely as possible. However, although statistics is highly useful in reducing the uncertainty in a response model, it does not eliminate all of the sources of the uncertainty. Therefore the resulting optimization model, constructed from the empirical data through careful adjustment and calibration using statistical methods, is not a perfect mirror of the true relationship; it is an approximation of the true mechanism in the underlying process or product. We have an opportunity in the optimization stage to address the uncertainty inherent to the statistical model to enhance the optimal solution.

In the context of quality engineering, response surface methodology (RSM) is a set of statistical and optimization techniques that are used sequentially to identify the optimal solution to a quality improvement problem. The iterative procedure in RSM includes performing an experiment in the region of the best known solution, fitting a response model to the experimental data, and optimizing the estimated response model. RSM has been widely used in quality engineering since the seminal work of George Box in the 1950s; for more details see *Box* and *Wilson* [12.5]. We give a brief introduction to RSM in Sect. 12.2 of this chapter.

In RSM, the optimization procedure is performed directly on the estimated response model, so it does not deliver a solution that minimizes the uncertainty in the model estimation process. This chapter is motivated by the work in *Xu* and *Albin* [12.6] and provides

an introduction into how we can use robust optimization methods to identify good solutions that maximize the performance of a process or product and, in the meantime, address the uncertainty inherent to a response model that is an approximation of the true relationship between the design parameters and the process/product performance. To make this idea clearer, consider the following function $y = f(\boldsymbol{x}, \boldsymbol{\beta})$, where y is the true process/product performance, $\boldsymbol{x}$ includes a set of design parameters and the function $f(x, \cdot)$ describes the true relationship between the design parameters in x and the process/product performance y. The vector $\boldsymbol{\beta}$ captures the important parameters in the function $f(x, \cdot)$.

If the function is a first-order polynomial

$$y = \sum_{i=1}^{n} \beta_i x_i \ ,$$

then $\boldsymbol{\beta}$ is a vector including all the coefficients $(\beta_1, \beta_2, \cdots \beta_n)'$. If the function is a second-order polynomial

$$y = \sum_{1 \le i \le j \le n} \beta_{ij} x_i x_j + \sum_{i=1}^{n} \beta_i x_i \ ,$$

then the vector $\boldsymbol{\beta}$ can be written as

$$(\beta_{11}, \beta_{12}, \cdots \beta_{nn}, \beta_1, \beta_2, \cdots \beta_n)' \ .$$

We note that the function $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in the coefficients in $\boldsymbol{\beta}$ when $f(\boldsymbol{x}, \boldsymbol{\beta})$ is a polynomial of x. This property plays an important role in the robust method introduced in this chapter.

The parameters in $\boldsymbol{\beta}$ are important in that they characterize how a process or product behaves. For example, let us consider a mixture design problem on glass/epoxy composites. We are interested in choosing the optimal mix of glass and epoxy to maximize the strength of the composites. Assume the relationship between the strength y and the fraction of glass (x_1) and epoxy (x_2) can be described by the response function $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$. The parameter β_1 (β_2) measures how the composite strength changes in response to the change in the fraction of glass (epoxy) while the parameter β_3 measures the glass–epoxy interaction effect on the composite strength.

Although the parameters in $\boldsymbol{\beta}$ are crucial in determining the behavior of a process or product, the true values for $\boldsymbol{\beta}$ are usually unknown to quality engineers. The only way to derive the values for $\boldsymbol{\beta}$ is by fitting a statistical model to the experimental data. Since the coefficients in $\boldsymbol{\beta}$ are estimated values, instead of writing $y = f(\boldsymbol{x}, \boldsymbol{\beta})$, we will use the notation $y = f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is estimated from historical or experimental data.

In quality engineering problems, we usually use the canonical optimization approach to determine the optimal solution. We first estimate the model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ from the experimental data and treat it as a true characterization of the underlying model. Then we solve for the optimal solution to the model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$. In the canonical approach, the point estimates in $\hat{\boldsymbol{\beta}}$ are regarded as a single representation of the true parameters in $\boldsymbol{\beta}$ and thus the optimization steps do not take into account the uncertainty about the estimate $\hat{\boldsymbol{\beta}}$. Although the canonical approach provides a simple, practical way to optimize the process/product performance, the solution obtained from the canonical approach may be far from optimal under the true performance response model.

Figure 12.1 illustrates the potential danger of the canonical approach when the performance response model is a second-order model. The dashed curve on the right represents the true, but unknown, model and the solid curve on the left the fitted model. If the goal is to minimize the performance response, the optimal value of the design variable is D_0 and the optimal performance response is 0. The canonical approach selects the value D_1 for the design variable, which results in the performance response P_1, well above the true optimal. Thus, even a slight deviation of the fitted model from the true model might result in unacceptable performance.

Section 12.2 in this chapter presents the robust optimization approach to address the pitfall illustrated in the above example. In contrast to the canonical approach, where uncertainty about the estimates $\hat{\boldsymbol{\beta}}$ is not explicitly addressed and only a single model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ is optimized, the robust approach considers a family of models and each model in the family is a possible representation of the true model. Robust and canonical optimization

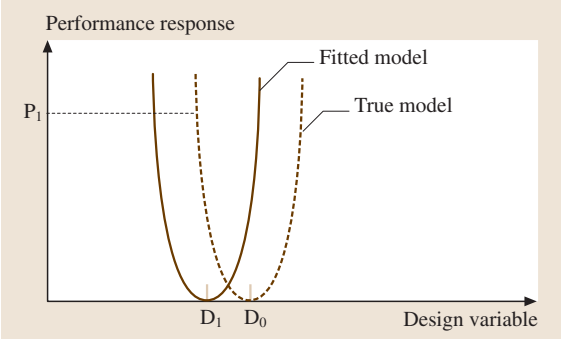


Fig. 12.1 Optimizing the estimated model yields performance response P_1, significantly higher than the true minimum 0

are compared using a Monte Carlo simulation example in Sect. 12.2, in the context of response surface methodology (RSM), a widely used method to optimize product/process.

The single estimated model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ in the canonical approach is the most likely representation of the true model $f(\boldsymbol{x}, \boldsymbol{\beta})$, while the robust approach incorporates more information by considering a family of models. Section 12.3 takes this a step further by combining each individual model in this family with a likelihood measure of how close it is to the true $f(\boldsymbol{x}, \boldsymbol{\beta})$. The improved approach presented in Sect. 12.3 is called the weighted robust optimization method and we prove that it provides a more effective solution to the estimated optimization model.

One of the greatest achievements in quality engineering in the last century is the robust design method that Taguchi proposed to minimize the expectation of Taguchi's loss function. The loss function is a measure of the deviation of the product performance from the desired target. The quality of a product can be measured by the loss function so a robust design problem can be reduced to an optimization problem whose objective is to minimize the expectation of the loss function by choosing optimal levels in design parameters.

To obtain the objective function in the robust design problem, designed experiments must be conducted, experimental data be collected and the loss function be fitted from the data. We are confronted with the same problems as discussed earlier on the uncertainty associated with the inference from the experimental data. Therefore robust optimization approach can be applied to identify a robust set of values for the design parameters. Section 12.4 discusses how we can leverage the robust optimization methods to better address Taguchi's robust design problems.

12.1 An Introduction to Response Surface Methodology

Response surface methodology (RSM) is a sequential approach and comprises iterative steps to conduct designed experiments, estimate the response model and derive the optimal solution in sequence. This section introduces the most essential steps in RSM and we refer the reader to *Box* and *Draper* [12.2], *Myers* and *Montgomery* [12.3], *Khuri* and *Cornell* [12.4] for a more comprehensive introduction to RSM.

We assume that, prior to running RSM, we have selected a list of significant factors that are the most important contributors to the response. Screening experiments such as fractional factorial designs and Plackett–Burman designs can be used to identify the important factors; see *Wu* and *Hamada* [12.7].

Let $\boldsymbol{x} = (x_1, x_2, \cdots x_k)$ denote the factors we have selected. We first run a first-order experiment such as 2^k factorial designs and fit a linear model to the experimental data. The next step is to choose the path of steepest ascent or steepest descent, run several experiments along the path and choose the one with the best performance response. We move the experimental region to the new location identified on the steepest ascent (descent) path and run the first-order experiments using the same steps above. We continue this process until no improvement is possible using first-order experiments. A second-order experiment, such as central composite designs, is conducted in order to describe the response surface better. We then solve a quadratic optimization model, obtain the solution and run confirmatory experiments to validate the optimal solution.

We use a paper helicopter example to illustrate the steps described above. The purpose of this exercise is to design a paper helicopter by choosing the optimal levels for rotor length/width, body length/width and other factors to maximize the flight time of the helicopter. Due to the convenience of the design, this exercise has been used in several institutions to teach design of experiments and RSM. We use the results presented in *Erhardt* and *Mai* [12.8] to demonstrate the basic steps in RSM. Another good source for the design of paper helicopter using RSM can be found in *Box* and *Liu* [12.9].

In *Erhardt* and *Mai* [12.8], there are eight factors that are likely to contribute to the flight time of the paper helicopter: rotor length, rotor width, body length, foot length, fold length, fold width, paper weight, and direction of fold. Screening experiments were conducted and the investigators found that two of the eight variables, rotor length and rotor width, are important in determining the flight time.

Erhardt and *Mai* [12.8] conducted a 2^2 factorial experiment with replicated runs and center points. The experimental data is shown in Table 12.1. The coded level 1 and -1 for rotor length stands for 11.5 and 5.5 cm, respectively. The coded level 1 and -1 for rotor width stands for 5 and 3 cm, respectively.

The first-order model fitted to the data in Table 12.1 is

$$\text{Flight time} = 11.1163 + 1.2881 \times \text{Rotor length} - 1.5081 \times \text{Rotor width}\,.$$

Therefore the path of steepest ascent is (1.2881, −1.5081) in coded level; in other words, for every one centimeter of increase in rotor length, rotor width should be decreased by

$$\frac{1.5081}{1.2881} \times \frac{1}{3} = 0.39\,\text{cm}\,.$$

The investigators conducted five experiments along the steepest ascent path and the experimental data is recorded in Table 12.2. The combination of rotor length and width that gives the longest flight time is 11.5 and 2.83 cm.

The investigators then conduct a central composite design (CCD) by adding experimental runs at axial points. Table 12.3 below contains the data from the CCD experiment. The center point of the CCD design is (11.5, 2.83), which is the solution obtained from the experimental runs on the steepest ascent path. One coded unit stands for 1 cm for rotor length and 0.39 cm for rotor width.

Table 12.1 2^2 factorial design for paper helicopter example

Coded level		Actual level (cm)		Flight time (seconds)			
Rotor length	Rotor width	Rotor length	Rotor width	Replicate 1	Replicate 2	Replicate 3	Replicate 4
1	1	11.5	5	10.02	9.94	9.95	9.93
1	-1	11.5	3	16.52	16.99	12.58	13.86
-1	1	5.5	5	10.20	9.26	8.20	9.92
-1	-1	5.5	3	10.24	9.11	11.31	10.94
0	0	8.5	4	11.67	10.74	9.83	

Table 12.2 Experiments along the path of steepest ascent

	Rotor length (cm)	Rotor width (cm)	Flight time (s)
Base	8.5	4	12.99
Path of steepest ascent Δ	1	-0.39	
Base + 1×Δ	9.5	3.61	15.22
Base + 2×Δ	10.5	3.22	16.34
Base + 3×Δ	11.5	2.83	18.78
Base + 4×Δ	12.5	2.44	17.39
Base + 5×Δ	13.5	2.05	7.24

Table 12.3 Central composite design for paper helicopter example

Coded level		Actual level (cm)		Flight time (s)
Rotor length	Rotor width	Rotor length	Rotor width	
1	1	12.5	3.22	13.53
1	-1	12.5	2.44	13.74
-1	1	10.5	3.22	15.48
-1	-1	10.5	2.44	13.65
$\sqrt{2}$	0	12.91	2.83	12.51
$-\sqrt{2}$	0	10.08	2.83	15.17
0	$\sqrt{2}$	11.5	3.38	14.86
0	$-\sqrt{2}$	11.5	2.28	11.85
0	0	11.5	2.83	17.38
0	0	11.5	2.83	16.35
0	0	11.5	2.83	16.41

The second-order model fitted to the data in Table 12.3 is given below:

$$\text{Flight time} = 16.713 - 0.702x_1 + 0.735x_2 - 1.311x_1^2 - 0.510x_1x_2 - 1.554x_2^2 ,$$

where x_1 stands for rotor length and $x_2$stands for rotor width. The optimal solution by maximizing this quadratic model is $(-0.32,\ 0.29)$ in coded units, or 11.18 cm for rotor length and 2.94 cm for rotor width.

The paper helicopter example presented in this section is a simplified version of how response surface methodology works to address quality improvement. A complicated real-world problem may require many more iterations in order to find an optimal solution and many of the technical details can be found in the references given in the beginning of this section.

12.2 Minimax Deviation Method to Derive Robust Optimal Solution

As we discussed in the introduction, the estimated model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ is a single representation of the true relationship between the response y and the predictor variables in $\boldsymbol{x}$, where $\hat{\boldsymbol{\beta}}$ is a point estimate and is derived from the sample data. The solution obtained by optimizing a single estimated model $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$ may not work well for the true model $f(\boldsymbol{x}, \boldsymbol{\beta})$. This section introduces the minimax deviation method to derive the robust solution when the experimental or historical data is available to estimate the optimization model.

One assumption we make here is that the vector $\boldsymbol{\beta}$ in $f(\boldsymbol{x}, \boldsymbol{\beta})$ contains the coefficients in the model and we assume that $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in the coefficients in $\boldsymbol{\beta}$. This assumption covers a wide range of applications since most of the models considered in quality engineering are derived using regression and the hypothetical model $f(\boldsymbol{x}, \boldsymbol{\beta})$ is always linear in regression coefficients even if the model itself is nonlinear in $\boldsymbol{x}$. For example, consider $f(\boldsymbol{x}, \boldsymbol{\beta}) = \beta_1 x^2 + \beta_2 x + \beta_3 \frac{1}{x}$, clearly $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in $(\beta_1, \beta_2, \beta_3)$, although it is not linear in $\boldsymbol{x}$.

12.2.1 Motivation of the Minimax Deviation Method

Consider two models in Fig. 12.2 where model 1 is $y = f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)})$ and model 2 is $y = f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)})$. If we assume that model 1 and model 2 are equally likely to be the true one, then how do we choose the value for x to minimize the response y in the true model? If the value at point A is chosen, there is a 50% chance that the response value y reaches its minimum if model 1 is the true model, while we are facing another 50% chance that the response value y is much worse when model 2 is the true model. A similar conclusion can be made if point B is chosen. Thus a rational decision maker will probably

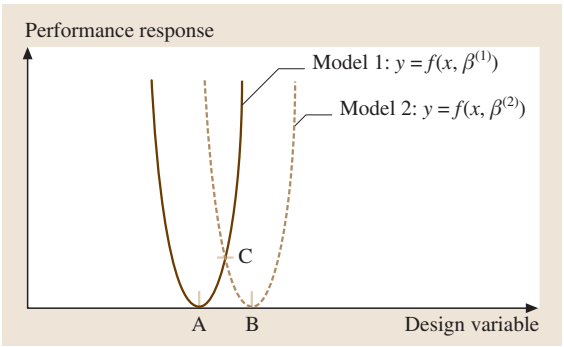


Fig. 12.2 Point C makes the response value close to the minimum whether model 1 or model 2 is the true model

choose point C such that the response value y will not be too far off from the minimum 0 whether model 1 or model 2 is the true one.

To formalize the reasoning, we use the following notation: let g_1 be the minimum value of $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)})$, and g_2 be the minimum value of $f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)})$. For the example in Fig. 12.2, $g_1$and g_2 are both zeros. Given that model 1 and model 2 are equally likely to be the true model, a rational decision maker wants to find an x such that, when the true model is model 1, the response value at $\boldsymbol{x}$, or $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)})$, is not too far from g_1; and when the true model is model 2, the response value at x, or $f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)})$, is not too far from g_2. In other words, we want to select x such that both $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)}) - g_1$ and $f(\boldsymbol{x}, \boldsymbol{\beta}^{(12.1)}) - g_2$ are as small as possible. Mathematically this is equivalent to the following problem.

Choose $\boldsymbol{x}$ to minimize

$$\text{Max}\left[f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)}) - g_1,\ f(\boldsymbol{x}, \boldsymbol{\beta}^{(2)}) - g_2 \right] . \qquad (12.1)$$

The difference $f(\boldsymbol{x}, \boldsymbol{\beta}^{(1)}) - g_1$ can be understood as the *regret* a rational decision maker will have if he

chooses this particular x when the true model is model 1, since g_1 is the minimum value the response can reach under model 1. Similarly the difference $f(\boldsymbol{x}, \boldsymbol{\beta}^{(2)}) - g_2$ is the *regret* a rational decision maker will have when the true model is model 2. Thus the aim of (12.1) is to choose an $\boldsymbol{x}$ to minimize the maximum regret over the two likely models.

12.2.2 Minimax Deviation Method when the Response Model Is Estimated from Data

Given the motivation in the previous section where the true model has two likely forms, we now consider the case where the response model is estimated from sample data; thus there are infinitely many forms that are likely the true model. Let the experimental data be $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2) \cdots (\boldsymbol{x}_n, y_n)$, where $\boldsymbol{x}_i$ contains predictor variables for the i^{th} observation and y_i is the corresponding response value. Suppose the true model is $y = f(\boldsymbol{x}, \boldsymbol{\beta})$where $\boldsymbol{\beta}$ contains the parameters we will fit using the experimental data. The estimate for $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}}$, can be derived using the MLE or LSapproach. The estimated model using the point estimate $\hat{\boldsymbol{\beta}}$, or $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$, is only one of the many possible forms for the true model $f(\boldsymbol{x}, \boldsymbol{\beta})$. Statistical inference provides ways to construct a confidence region, rather than a single-point estimate, to cover the possible value for the true $\boldsymbol{\beta}$. Let us denote a confidence region for $\boldsymbol{\beta}$ by B; thus any model $f(\boldsymbol{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta} \in B$, represents a likely true model.

Figure 12.3 illustrates how robust optimization works by incorporating all of the estimates in the confidence region. The rectangle in Fig. 12.3 is the confidence region for $\boldsymbol{\beta}$ derived from the sample data, and the center point of the rectangle is the point estimate $\hat{\boldsymbol{\beta}}$. The usual canonical approach optimizes only a single model corresponding to the point estimate, or $f(\boldsymbol{x}, \hat{\boldsymbol{\beta}})$. In contrast, robust optimization considers all of the possible estimates in the confidence region, so it optimizes all of the likely models $f(\boldsymbol{x}, \boldsymbol{\beta})$ whose $\boldsymbol{\beta}$ is in the rectangle.

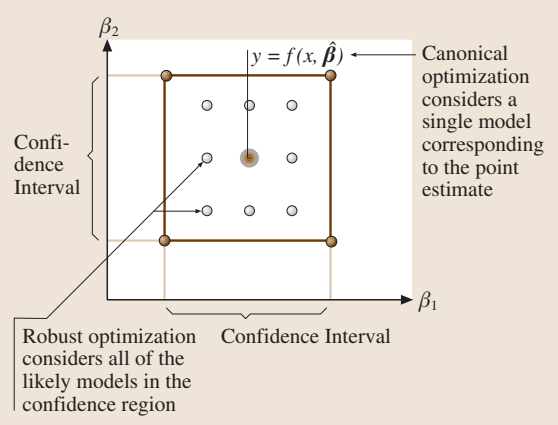


Fig. 12.3 Canonical optimization considers a single model while robust optimization considers all of the models with estimates in the confidence region

We now use the minimax deviation method in Sect. 12.2.1 to derive the robust solution where all of the likely models with estimates in the confidence region are considered. Suppose our goal is to minimize $f(\boldsymbol{x}, \boldsymbol{\beta})$ and the confidence region for $\boldsymbol{\beta}$ is B. The minimax deviation method can be formulated in the following equations:

$$\text{Min}_{\boldsymbol{x}} \text{Max}_{\boldsymbol{\beta} \in B} \left[f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta}) \right] , \tag{12.2}$$

where

$$g(\boldsymbol{\beta}) = \text{Min}_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\beta}), \text{ for any} \boldsymbol{\beta} \in B .$$

The interpretation of the minimax deviation method in (12.2) is similar to that given in Sect. 12.2.1. The difference $f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta})$ is the regret incurred by choosing a particular $\boldsymbol{x}$ if the true coefficients in the model are $\boldsymbol{\beta}$. However the true values for $\boldsymbol{\beta}$ are unknown and they are likely at any point in the confidence region B. So $\text{Max}_{\boldsymbol{\beta} \in B} [f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta})]$ stands for the maximum regret over the confidence region. We solve for the robust solution for $\boldsymbol{x}$ by minimizing the maximum regret over B.

The minimax deviation model in (12.2) is equivalent to the following mathematical program as in reference [12.10]

$$\begin{aligned} &\text{Min}(z) , \\ &f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta}) \le z, \quad \forall \boldsymbol{\beta} \in B , \\ &g(\boldsymbol{\beta}) = \text{Min}_{\boldsymbol{x}} \left[f(\boldsymbol{x}, \boldsymbol{\beta}) \right] . \end{aligned} \tag{12.3}$$

The number of decision variables in this statement is finite while the number of constraints is infinite because every constraint corresponds to a point in the confidence region, or the rectangle in Fig. 12.3. Therefore the program in (12.3) is semi-infinite.

As illustrated in Fig. 12.3, we assume the confidence region can be constructed as a polytope. With this assumption, we have the following reduction theorem.

Reduction theorem. If B is a polytope and $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in $\boldsymbol{\beta}$ then

$$\begin{aligned} &\text{Min}_{\boldsymbol{x}} \text{Max}_{\boldsymbol{\beta} \in B} \left[f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta}) \right] \\ &= \text{Min}_{\boldsymbol{x}} \text{Max}_i \left[f(\boldsymbol{x}, \boldsymbol{\beta}^{\text{i}}) - g(\boldsymbol{\beta}^{\text{i}}) \right] , \end{aligned}$$

where $\boldsymbol{\beta}^1, \boldsymbol{\beta}^2 \cdots \boldsymbol{\beta}^m$ are the extreme points of B.

The reduction theorem says that the minimization of the maximum regret over the entire confidence region is equivalent to the minimization of the maximum regret over the extreme points of the confidence region. Figure 12.4 illustrates the use of the reduction theorem that reduces the original semi-infinite program in (12.3) to a finite program. The proof of the reduction theorem can be found in *Xu* and *Albin* [12.6].

12.2.3 Construction of the Confidence Region

One of the assumptions of the reduction theorem is that the confidence region for $\boldsymbol{\beta}$ is a polytope. This section introduces how we can construct a confidence region as a polytope.

A simple and straightforward way to construct a confidence polytope is to use simultaneous confidence intervals (*Miller* [12.11]). Suppose $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_p)$ and we want to construct a confidence polytope with a confidence level of $(1-\alpha)\times 100\%$ or more. First we construct a $(1-\alpha/p)\times 100\%$ confidence interval for each of the p coefficients in $\boldsymbol{\beta}$. Specifically, let I_i be the $(1-\alpha/p)\times 100\%$ confidence interval for $\boldsymbol{\beta}_i$, or equivalently, $P(\boldsymbol{\beta}_i \in I_i) = 1-\alpha/p$. Thus the simultaneous confidence intervals is the Cartesian product $B = I_1 \times I_2 \times \cdots \times I_p$. Using Bonferroni's inequality, we have

$$\begin{aligned} P(\boldsymbol{\beta} \in B) &= P\left(\boldsymbol{\beta}_1 \in I_1, \boldsymbol{\beta}_2 \in I_2, \cdots \boldsymbol{\beta}_p \in I_p\right) \\ &\geq 1 - \sum_{i=1}^{p} P(\boldsymbol{\beta}_i \notin I_i) \\ &= 1 - p\times\alpha/p = 1-\alpha\,. \end{aligned}$$

Therefore the confidence level of the simultaneous confidence intervals B is at least $(1-\alpha)\times 100\%$. Figure 12.5 illustrates the simultaneous confidence intervals in a two-dimensional space. Suppose the ellipsoid in the left panel of Fig. 12.5 is a 90% confidence region for $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$. To construct simultaneous confidence intervals, we first identify the 95% confidence interval I_1 for $\boldsymbol{\beta}_1$, and the 95% confidence interval I_2 for $\boldsymbol{\beta}_2$; thus the rectangle $I_1 \times I_2$ is a confidence polytope for $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ with a confidence level of at least 90%. However, we know from Fig. 12.5 that the rectangle does not cover the 90% confidence ellipsoid very tightly, so the simultaneous confidence intervals are not the smallest confidence polytope at a certain confidence level. Clearly a better way to construct a more efficient confidence polytope is to find a rectangle that circumscribes the ellipsoid, such as that in the right panel of Fig. 12.5.

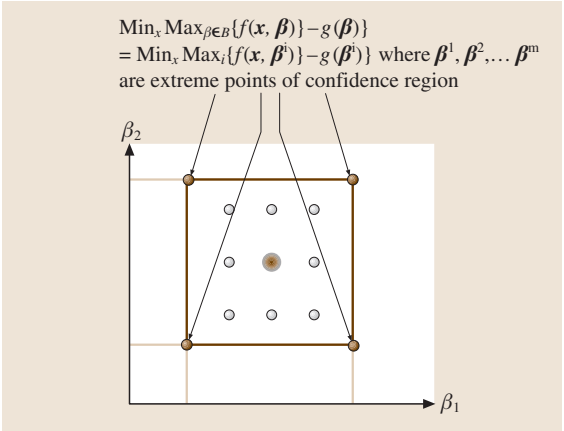


Fig. 12.4 The reduction theorem reduces the semi-infinite program over the entire confidence region to a finite program over the set of extreme points of the confidence region

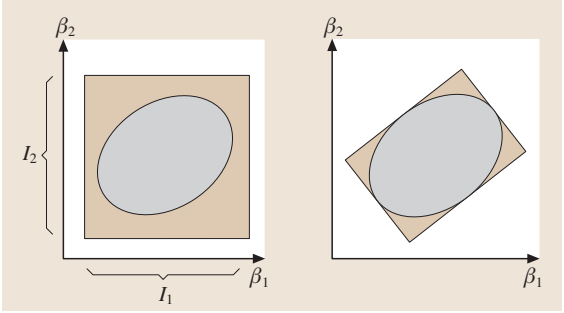


Fig. 12.5 Simultaneous confidence intervals are not the most efficient confidence polytope

We now present a transformation method to construct a tighter confidence polytope, which proves very effective to enhance robust optimization performance. Let $\boldsymbol{X}$ be a matrix with each row representing the observed values for the predictor variables in $\boldsymbol{x}$, and let $\boldsymbol{Y}$ be a vector with each element being the observed response value y.

From regression analysis, the $(1-\alpha)\times 100\%$ confidence region for $\boldsymbol{\beta}$ is an ellipsoid described as

$$\begin{aligned} &(1-\alpha)\times 100\% \text{ confidence region} \\ &= \left(\boldsymbol{\beta} \middle| \frac{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'(\mathbf{X}'\mathbf{X})(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{p\mathrm{MSE}} \leq F_{p,n-p,\alpha}\right), \end{aligned} \quad (12.4)$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ is the point estimator, p is the number of parameters we need to estimate in the response model, n is the total number of observations

we have in the sample data, $\text{MSE} = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})/n - p$ is the mean squared error, and $F_{p,n-p,\alpha}$ is the $(1-\alpha)\times 100$ percentile point for the F distribution with p and $(n-p)$ degrees of freedom. Details about (12.4) can be found in *Myers* [12.12].

We use Fig. 12.6 to illustrate the motivation for the transformation method to construct the confidence polytope in two dimensions. The ellipsoid in the left-hand picture of Fig. 12.6 is the $(1-\alpha)\times 100\%$ confidence region in (12.4). We want to find a polytope to cover the confidence ellipsoid more tightly. One such choice is to identify a rectangle with sides parallel to the major and minor axes of the ellipsoid, such as the one with vertices $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2$, $\boldsymbol{\beta}^3$ and $\boldsymbol{\beta}^4$ in Fig. 12.6.

It is hard to identify these extreme points $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2$, $\boldsymbol{\beta}^3$ and $\boldsymbol{\beta}^4$ directly in the original coordinate system $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$. However, by choosing appropriate algebraic transformation, the coordinate system $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ can be transformed into the coordinate system (z_1, z_2), where the ellipsoid is converted to a unit ball in the right-hand picture of Fig. 12.6. In the coordinate system (z_1, z_2), it is easy to find a hypercube, with extreme points z^1, z^2, z^3 and z^4, to cover this ball tightly. We then map these extreme points back to the extreme points in $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ to obtain $\boldsymbol{\beta}^1$, $\boldsymbol{\beta}^2$, $\boldsymbol{\beta}^3$ and $\boldsymbol{\beta}^4$.

To achieve this idea, we define the following transformation $\boldsymbol{\beta} \mapsto \mathrm{z}$:

$$\mathbf{z} = \boldsymbol{\Gamma}\,(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\,, \text{ where } \Gamma = \frac{(\mathbf{X}'\mathbf{X})^{1/2}}{\sqrt{p \times \text{MSE} \times F_{p,n-p,\alpha}}}\,.$$

Through this transformation, the confidence ellipsoid in (12.4) in the coordinate system $\boldsymbol{\beta}$ can be converted into a unit ball in the coordinate system z: $\left(\mathrm{z}|\mathrm{z}'\mathrm{z} \le 1\right)$. It is easy to know that the hypercube covering the unit ball has extreme points $\mathrm{z}^i = (z_1, z_2, \cdots, z_p)$, where $z_j = \pm 1$, $j = 1, 2, \cdots, p$. By mapping these points back to the coordinate system $\boldsymbol{\beta}$, we can construct a confidence polytope with extreme points as follows:

$$\boldsymbol{\beta}^i = \hat{\boldsymbol{\beta}} + \boldsymbol{\Gamma}^{-1}\mathbf{z}^i, \text{ where } \Gamma = \frac{(\mathbf{X}'\mathbf{X})^{1/2}}{\sqrt{p \times \text{MSE} \times F_{p,n-p,\alpha}}}\,. \tag{12.5}$$

Thus the robust optimization model in (12.3) can be written as

$$\begin{aligned} &\text{Min}(z)\,, \\ &f(\boldsymbol{x}, \boldsymbol{\beta}^{\mathrm{i}}) - g(\boldsymbol{\beta}^{\mathrm{i}}) \le z\,, \\ &g(\boldsymbol{\beta}^{\mathrm{i}}) = \text{Min}_{\boldsymbol{x}}\, f(\boldsymbol{x}, \boldsymbol{\beta}^{\mathrm{i}})\,, \end{aligned} \tag{12.6}$$

where $\boldsymbol{\beta}^{\mathrm{i}}$ is given in (12.5)

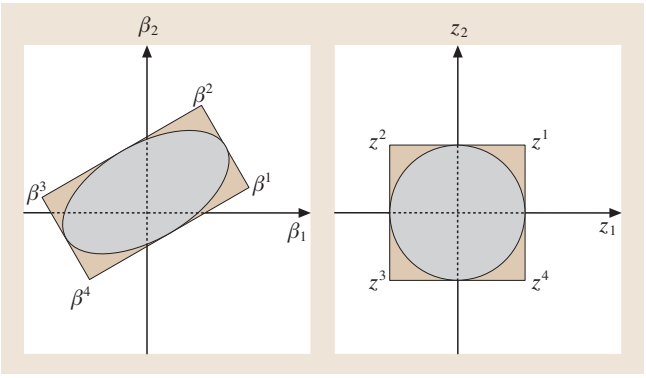


Fig. 12.6 Illustration of the transformation method to construct a confidence polytope

12.2.4 Monte Carlo Simulation to Compare Robust and Canonical Optimization

This section compares the performance of robust optimization and canonical optimization using Monte Carlo simulation on a hypothetical response model. Much of the material is from *Xu* and *Albin* [12.6] and *Xu* [12.13]. Suppose the true function relating performance response $y$and design variables x_1 and x_2 is the quadratic function

$$y = 0.5x_1^2 - x_1x_2 + x_2^2 - 2x_1 - 6x_2\,. \tag{12.7}$$

The objective is to identify x_1 and x_2 to minimize y with the constraints that $x_1 + x_2 \le 6$, $x_1 \ge 0$, and $x_2 \ge 0$. If the response model in (12.7) is known, the true optimal solution can be easily identified: $x_1 = 2.8$, $x_2 = 3.2$, yielding the optimal value $y = -19.6$.

Now suppose that the objective function is not known. We could perform a designed experiment to estimate the performance response function. Since we seek a second-order function we would perform a 3^2 factorial design with three levels for x_1 and three levels for x_2, resulting in a total of nine different combinations of $x_1$and x_2. The possible experimental values are -1, 0 and 1 for $x_1$and -1, 0, and 1 for x_2.

Instead of performing the experiment in a laboratory, we use Monte Carlo simulation, where the response y is produced by generating responses equal to the underlying response function in (12.7) plus noise ε:

$$\begin{aligned} &y = 0.5x_1^2 - x_1x_2 + x_2^2 - 2x_1 - 6x_2 + \varepsilon \\ &\quad \text{and } \varepsilon \sim N(0, \sigma^2)\,. \end{aligned} \tag{12.8}$$

Once the experiment has been run, we fit coefficients to the data by ordinary least-square regression and then optimize using the robust and canonical approaches, respectively.

The solutions obtained from the two approaches are inserted into (12.7) to determine the resulting performance response values and we compare these to determine which is closer to the true optimal.

We perform the above experiment and subsequent optimizations 100 times for each of the following degrees of experimental noise; that is, the noise term ε in (12.8) has standard deviation, σ, equal to 0.5, 1, 2, 3, or 4.

Thus we have 100 objective values for the canonical approach and 100 objective values for the robust approach for each value of σ. Table 12.4 gives the means and standard deviations of these performance responses using the canonical approach, the robust approach with simultaneous confidence intervals, and the robust approach with transformation method.

Table 12.4 shows that, when the experimental noise is small (σ= 0.5), yielding a relatively accurate point estimate of $\boldsymbol{\beta}$, the objective values given by the canonical approach are slightly closer to those given by the robust optimization approach. However, when the experimental noise is large (σ= 1,2,3,4), yielding a relatively inaccurate point estimate of $\boldsymbol{\beta}$, the robust approach yields results much closer to the true optimal than the canonical approach. We also notice that the robust approach using transformation method to construct the confidence polytope gives better results than the method using the simultaneous confidence intervals.

12.3 Weighted Robust Optimization

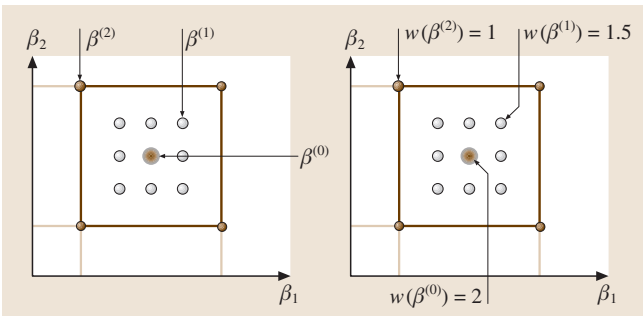


Fig. 12.7 Weighted robust optimization assigns weights to every point in the confidence region to reflect the likelihood of that point being close to the true $\boldsymbol{\beta}$

As we discussed earlier, robust optimization minimizes the maximum regret over a confidence region for the coefficients in the response model. Recall that the robust optimization is written as follows:

$$\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{\boldsymbol{\beta}\in B}\left[f(\boldsymbol{x},\boldsymbol{\beta})-g(\boldsymbol{\beta})\right]\ ,$$

where $g(\boldsymbol{\beta}) = \mathrm{Min}_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{\beta})$, for any $\boldsymbol{\beta}\in B$.

An implicit assumption in the minimax regret equation above is that all of the points in the confidence region B are treated with equal importance. For example, consider the three points $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\beta}^{(1)}$, and $\boldsymbol{\beta}^{(2)}$ in the left-hand picture of Fig. 12.7, the *regrets* we have at the three points by choosing $\boldsymbol{x}$ are $f(\boldsymbol{x},\boldsymbol{\beta}^{(0)})-g(\boldsymbol{\beta}^{(0)})$, $f(\boldsymbol{x},\boldsymbol{\beta}^{(1)})-g(\boldsymbol{\beta}^{(1)})$ and $f(\boldsymbol{x},\boldsymbol{\beta}^{(2)})-g(\boldsymbol{\beta}^{(2)})$, respectively. However, we know from statistical inference that the center point $\boldsymbol{\beta}^{(0)}$ is more likely close to be the true $\boldsymbol{\beta}$ than $\boldsymbol{\beta}^{(1)}$, and $\boldsymbol{\beta}^{(1)}$ is more likely to be close to the true $\boldsymbol{\beta}$ than$\boldsymbol{\beta}^{(2)}$, so in the *regret* calculation, weights can be assigned to each point in the confidence region to measure how likely that point is to be close to the true $\boldsymbol{\beta}$.

In the right-hand picture of Fig. 12.7, the weights for the three points are $w\left(\boldsymbol{\beta}^{0}\right)=2$, $w\left(\boldsymbol{\beta}^{(1)}\right)=1.5$, and $w\left(\boldsymbol{\beta}^{(2)}\right)=1$, so the *regrets* at these three points can be defined as $2\left[f\left(\boldsymbol{x},\boldsymbol{\beta}^{(0)}\right)-g\left(\boldsymbol{\beta}^{(0)}\right)\right]$, $1.5\left[f\left(\boldsymbol{x},\boldsymbol{\beta}^{(1)}\right)-g\left(\boldsymbol{\beta}^{(1)}\right)\right]$ and $\left[f\left(\boldsymbol{x},\boldsymbol{\beta}^{(2)}\right)-g\left(\boldsymbol{\beta}^{(2)}\right)\right]$. In general, the weighted robust optimization can be written as

$$\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{\boldsymbol{\beta}\in B}\left[f(\boldsymbol{x},\boldsymbol{\beta})-g(\boldsymbol{\beta})\right]w(\boldsymbol{\beta})\ , \qquad (12.9)$$

where $w(\boldsymbol{\beta})$ is the weight assigned to the point $\boldsymbol{\beta}$ in the confidence region. So the aim of the weighted robust optimization in (12.9) is to minimize the maximum weighted regret over the confidence region. The center point of the confidence region should be assigned the largest weight since it is most likely to be close to the true $\boldsymbol{\beta}$. On the other hand, the extreme points of the confidence region should be assigned the smallest weights.

We now consider two choices of the weight function $w(\boldsymbol{\beta})$. Let $\boldsymbol{\beta}^{(0)}$ be the center point of the confidence region; let $\boldsymbol{\beta}^{(+)}$ be an extreme point with the largest distance to $\boldsymbol{\beta}^{(0)}$. In the first version of weight function, we treat the point $\boldsymbol{\beta}^{(0)}$ as twice as important as $\boldsymbol{\beta}^{(+)}$. In other words, we assign weight 1 to the extreme point $\boldsymbol{\beta}^{(+)}$ and the weight for the center point $\boldsymbol{\beta}^{(0)}$ is 2. The weight for any other point $\boldsymbol{\beta}$ is between 1 and 2 and decreases linearly with its distance from the center point $\boldsymbol{\beta}^{(0)}$. This linear-distance-based weight function can be

written in the following way:

$$w(\boldsymbol{\beta}) = 2 - \frac{||\boldsymbol{\beta} - \boldsymbol{\beta}^{(0)}||}{||\boldsymbol{\beta}^{(+)} - \boldsymbol{\beta}^{(0)}||} \,. \tag{12.10}$$

We now discuss the second version of weight function. Let $\boldsymbol{x}^{(\mathrm{i})}$and y_i, i = 1, 2, $\cdots$, n, be the observation for the predictors and response value. For any estimator $\boldsymbol{\beta}$ in the confidence region, the sum of squared errors (SSE) $\sum_{i=1}^{n}[\,\boldsymbol{y}_i - f(\boldsymbol{x}^{(\mathrm{i})}, \boldsymbol{\beta})]^2$ can be viewed as an indirect measure of how close the estimator $\boldsymbol{\beta}$ is to the true coefficients. So we take the reciprocal of the SSE as the weight function, or

$$w(\boldsymbol{\beta}) = \frac{1}{\sum_{i=1}^{n}[\,\boldsymbol{y}_i - f(\boldsymbol{x}^{(\mathrm{i})}, \boldsymbol{\beta})]^2} \,. \tag{12.11}$$

To compare the performance of robust optimization and weighted robust optimization, we use the same underlying response model in (12.7) to generate the experimental data and then apply the two approaches to derive the robust solutions. Table 12.5 contains the means and standard deviations of the performance responses obtained by the canonical, robust and weighted robust optimization approaches.

It is clear that the performance of the weighted robust optimization dominates that of the standard robust optimization. We further note that the weight function in (12.10) performs better than the weight function in (12.11) when the experimental noise is large ($\sigma = 1, 2, 3, 4$).

Although the weighted robust optimization gives better results, computationally it is much harder and more challenging. Unfortunately, weighting the points in the confidence region, using weight functions $w(\boldsymbol{\beta})$ in (12.10) or (12.11), results in an optimization problem in (12.9) with an objective function that is not linear in $\boldsymbol{\beta}$. Consequently, the reduction theorem is no longer applicable to (12.9) to reduce the optimization problem to a finite program. Therefore a numerical algorithm has to be designed to solve the weighted robust optimization problem in (12.9).

For simplicity, let $F(\boldsymbol{x}, \boldsymbol{\beta}) = [\,f(\boldsymbol{x}, \boldsymbol{\beta}) - g(\boldsymbol{\beta})]\, w(\boldsymbol{\beta})$. Thus we can write the weighted robust optimization problem as follows

$$\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{\boldsymbol{\beta}\in B} F(\boldsymbol{x}, \boldsymbol{\beta}) \tag{12.12}$$

or equivalently,

$$\begin{aligned} &\mathrm{Min}_{\boldsymbol{x}}\{\xi\}\,, \\ &\text{s.t. x} \in X\,, \\ &F(\boldsymbol{x}, \boldsymbol{\beta}) \le \xi, \forall \boldsymbol{\beta} \in B\,. \end{aligned} \tag{12.13}$$

We use the Shimizu–Aiyoshi relaxation algorithm to solve (12.13). For a rigorous treatment of this relaxation algorithm, see *Shimizu* and *Aiyoshi* [12.10]. The main steps in this algorithm are given as follows:

Step 1: choose any initial point $\boldsymbol{\beta}^{(1)}$. Set $k = 1$.

Step 2: solve the following relaxed problem of (12.13):

$$\begin{aligned} &\mathrm{Min}_{\boldsymbol{x}}\{\xi\} \\ &\text{s.t.}\, \boldsymbol{x} \in X \\ &F(\boldsymbol{x}, \boldsymbol{\beta}^{(i)}) \le \xi, i = 1, 2, \cdots, k \end{aligned} \tag{12.14}$$

Obtain an optimal solution $(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}^{(k)})$ for (12.14). The $\boldsymbol{\xi}^{(k)}$ is also the optimal value for (12.14). We note that $\boldsymbol{\xi}^{(k)}$ is a lower bound on the optimal value for (12.13).

Step 3: solve the maximization problem:

$$\mathrm{Max}_{\boldsymbol{\beta}\in B} F(\boldsymbol{x}^{(k)}, \boldsymbol{\beta})\,. \tag{12.15}$$

Obtain an optimal solution $\boldsymbol{\beta}^{(k+1)}$ and the maximal value $\phi(\boldsymbol{x}^{(k)}) = F(\boldsymbol{x}^{(k)}, \boldsymbol{\beta}^{(k+1)})$. We note that $\phi(\boldsymbol{x}^{(k)})$ is an upper bound on the optimal value of (12.12) or (12.13).

Step 4: If $\phi(\boldsymbol{x}^{(k)}) - \boldsymbol{\xi}^{(k)} < \varepsilon$, terminate and report the solution $\boldsymbol{x}^{(k)}$; otherwise, set $k = k$+1 and go back to step 2.

We now introduce the method for solving the optimization problems (12.14) and (12.15). First, we address (12.14). If the response model $f(\boldsymbol{x}, \boldsymbol{\beta})$ is linear in $\boldsymbol{x}$, then $F(\boldsymbol{x}, \boldsymbol{\beta})$ is also linear in $\boldsymbol{x}$; thus (12.14) is a linear programming problem if we assume that the feasible region $\boldsymbol{X}$ contains only linear constraints. If the response model $f(\boldsymbol{x}, \boldsymbol{\beta})$ is quadratic in $\boldsymbol{x}$, then $F(\boldsymbol{x}, \boldsymbol{\beta})$ is also quadratic in $\boldsymbol{x}$; thus (12.14) is a quadratically constrained quadratic programming problem.

Quadratically constrained quadratic programming (QCQP) is a very challenging problem. One efficient way to solve QCQP is to approximate it by a semidefinite program (SDP) and the solution for SDP usually provides a very tight bound on the optimal value of the QCQP. There exist very efficient and powerful methods to solve SDP and numerous software packages have been developed. After an approximate solution is obtained from SDP, we then use a randomized algorithm to search for a good solution for the original QCQP. For a comprehensive introduction to SDP, see *Vandenberghe* and *Boyd* [12.14]; for the connection between QCQP and SDP, see *Alizadeh* and *Schmieta* [12.15], and *Frazzoli* [12.16]; for software packages to solve SDP, see *Alizadeh* et al. [12.17], and *Sturm* [12.18].

We finally comment on the optimization problem (12.15). The objective function in (12.15) is

$F(\boldsymbol{x}^k, \boldsymbol{\beta}) = \{f(\boldsymbol{x}^k, \boldsymbol{\beta}) - g(\boldsymbol{\beta})\}w(\boldsymbol{\beta})$. We note that the function $g(\boldsymbol{\beta})$ has no closed-form expression since it is the minimal value of $f(\boldsymbol{x}, \boldsymbol{\beta})$. Here we use a powerful global optimization algorithm, called DIRECT, to solve (12.15). The DIRECT algorithm was proposed by *Jones* et al. [12.19]. There are several advantages of using DIRECT: it is a global optimization algorithm and has a very good balance between global searching and local searching, and it converges quite quickly; it does not need derivative information on the objective function. For software on DIRECT, see *Bjorkman* and *Holmstrom* [12.20].

12.4 The Application of Robust Optimization in Parameter Design

This section applies robust optimization to solve Taguchi's parameter design problem. The aim of parameter design is to choose optimal levels for the control factors to reduce the performance variation as well as to make the response close to the target. Section 12.4.1 introduces both traditional and more recent approaches to handling parameter design problems. Section 12.4.2 discusses how to use robust optimization to identify a robust solution for control factors when the response model is estimated from experimental data. Section 12.4.3 presents the robust optimization method to solve parameter design problem when the experimental data is from a fractional factorial design and some effects are aliased.

12.4.1 Response Model Approach to Parameter Design Problems

Parameter design was promoted by Genichi Taguchi in the 1950s and has since been widely used in quality engineering (*Taguchi* [12.21]). In parameter design, there are two sets of variables: control factors and noise variables. Control factors are those variables that can be set at fixed levels in the production stage; noise variables are those variables that we cannot control such as environmental conditions and material properties, and are hence assumed random in the production stage. The performance response is affected by both control factors and noise variables.

The contribution of Taguchi, among many others, is to recognize that interaction often exists between control factors and noise variables. Hence, appropriate levels of control factors can be selected to reduce the impact of noise variables on the performance response.

Taguchi proposed a set of methodologies, including inner–outer array design and signal-to-noise ratio (SNR), to identify optimal levels of control factors. *Welch* et al. [12.22] and *Shoemaker* et al. [12.23] have proposed the response model formulation, a more statistically sound method, to deal with parameter design problems.

In the response model approach, we first conduct experiments at appropriate levels of control factors and noise variables. Then we can fit the following model to relate the performance response to both control factors and noise variables:

$$\begin{aligned} y &= f(\boldsymbol{x}; \alpha, \gamma, \mu, A, \Delta) \\ &= \mu + \tfrac{1}{2}\mathrm{x}'Ax + \alpha' x + x'\Delta \mathrm{z} + \gamma' z + \varepsilon\,, \end{aligned} \tag{12.16}$$

where $\boldsymbol{x}$ represents the control factors and $\boldsymbol{z}$ the noise variables. Equation (12.16) includes first- and second-order terms in control factors, a first-order term in noise variables and an interaction term between control factors and noise variables. The noise variables $\boldsymbol{z}$ have a normal distribution with mean $\boldsymbol{0}$ and variance Σ_z, or $z \sim N(0, \Sigma_z)$. The ε term incorporates unidentified noise other than $\boldsymbol{z}$.

The difference between the response model in (12.16) and the response model $f(\boldsymbol{x}, \boldsymbol{\beta})$ in the previous sections is that the former divides the noise into $\boldsymbol{z}$ and ε and introduces a first-order term and an interaction term related to $\boldsymbol{z}$ while the latter has the noise only in ε. We further note that the coefficients $(\alpha, \gamma, \mu, A, \Delta)$ in (12.16) are estimated from designed experiments. More details about (12.16) can be found in *Myers* and *Montgomery* [12.3].

From (12.16), it is easy to derive the expected value and standard deviation of the response value y

$$\begin{aligned} E(y) &= \mu + \tfrac{1}{2}x'Ax + \alpha' x\,, \\ \mathrm{Var}(y) &= x'\Delta\Sigma_z\Delta' x + \gamma'\Sigma_z\gamma + \sigma_\varepsilon^2\,. \end{aligned}$$

Suppose our goal is to choose control factors $\boldsymbol{x}$ such that the response y is as close as possible to a target t. In Taguchi's parameter design problem, the criterion to identify optimal levels for control factors $\boldsymbol{x}$ is to minimize the following expected squared loss:

$$\begin{aligned} L(\boldsymbol{x}; \alpha, \gamma, \mu, A, \Delta) &= [E(y) - t]^2 + \mathrm{Var}(y) \\ &= \left(\mu + \tfrac{1}{2}x'Ax + \alpha' x - t\right)^2 \\ &\quad + x'\Delta\Sigma_z\Delta' x + \gamma'\Sigma_z\gamma + \sigma_\varepsilon^2\,. \end{aligned}$$

12.4.2 Identification of Control Factors in Parameter Design by Robust Optimization

Since the true values for the coefficients $(\alpha, \gamma, \mu, A, \Delta)$ in (12.16) are unknown and they are estimated from data, we use robust optimization to derive a robust solution $\boldsymbol{x}$ that is resistant to the estimation error. First we use the same method as in Sect. 12.2.3 to construct a confidence region B for the coefficients $(\alpha, \gamma, \mu, A, \Delta)$ and then we solve the following minimax deviation model:

$$\begin{aligned}
&\mathrm{Min}_{\boldsymbol{x}}\mathrm{Max}_{(\alpha,\gamma,\mu,A,\Delta)\in B} L(\boldsymbol{x};\alpha,\gamma,\mu,A,\Delta)\\
&-g(\alpha,\gamma,\mu,A,\Delta)\,,\\
&g(\alpha,\gamma,\mu,A,\Delta)=\mathrm{Min}_{\boldsymbol{x}}L(\boldsymbol{x};\alpha,\gamma,\mu,A,\Delta)\,,\\
&L(\boldsymbol{x};\alpha,\gamma,\mu,A,\Delta)=\left(\mu+\tfrac{1}{2}x'Ax+\alpha'x-t\right)^2\\
&+x'\Delta\Sigma_z\Delta'x+\gamma'\Sigma_z\gamma+\sigma_\varepsilon^2\,. \qquad (12.17)
\end{aligned}$$

Note that the model (12.17) is not linear in the coefficients $(\alpha, \gamma, \mu, A, \Delta)$, so we have to resort to a numerical optimization algorithm to solve it.

We use the following example from *Xu* [12.13] to show the application of robust optimization to the parameter design problem. Suppose there are two control factors A and B and one noise variable C. The underlying relationship between the performance response $y$and control/noise factors A, B and C is

$$y=3A+2B+0.15C+0.5AC-BC+6+\varepsilon\,, \qquad (12.18)$$

where $\varepsilon\sim N(0,1)$. We further assume that the variance of the noise factor C is $\sigma_C=1$. Our goal is to choose the optimal levels of (A, B) over the feasible region $\{(A, B)|-1\le A, B\le 1\}$ to make the response y close to the target $t=7.5$. We notice that the response values have a minimum squared loss of 1 when the control factors $(A, B)=(0.3, 0.3)$.

Assume we do not know the true model (12.18), so we have first to fit a response model $y=\mu+\alpha_1A+\alpha_2B+\gamma_1C+\delta_1AC+\delta_2BC$, where $(\mu, \alpha_1, \alpha_2, \gamma_1, \delta_1, \delta_2)$ are the coefficients we will estimate. Suppose we perform a full 2^3 factorial design with the design matrix and the observed responses as follows.

Using the experimental data from the factorial design in Table 12.3, we first construct the confidence region B for the coefficients $(\mu, \alpha_1, \alpha_2, \gamma_1, \delta_1, \delta_2)$ in the response model. The squared loss for the response value y is $L=[E(y)-t]^2+\mathrm{Var}(y)=(\mu+\alpha_1A+\alpha_2B-t)^2+(\gamma_1+\delta_1A+\delta_2B)^2\sigma_C^2+\sigma_\varepsilon^2$. By substituting $t=7.5$ and $\sigma_C=\sigma_\varepsilon=1$, we have $L=(\mu+\alpha_1A+\alpha_2B-7.5)^2+(\gamma_1+\delta_1A+\delta_2B)^2+1$. Therefore we can write the robust optimization model as follows:

$$\begin{aligned}
&\mathrm{Min}_{\{-1\le A,B\le 1\}}\mathrm{Max}_{(\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\in B}\\
&L(A,B;\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\\
&-g(\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\,,\\
&g(\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\\
&=\mathrm{Min}_{(A,B)}L(A,B;\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\,,\\
&L(A,B;\mu,\alpha_1,\alpha_2,\gamma_1,\delta_1,\delta_2)\\
&=(\mu+\alpha_1A+\alpha_2B-7.5)^2\\
&+(\gamma_1+\delta_1A+\delta_2B)^2+1\,. \qquad (12.19)
\end{aligned}$$

By solving the optimization problem in (12.19), we can obtain the robust solution $(A, B) = (0.35, 0.22)$. If this solution is applied to the underlying model (12.18), the response values would have an expected squared loss 1.01, which is quite close to the true minimum 1. To be complete, we also present the results obtained by canonical optimization. The canonical solution is $(A, B) = (0.29, 0.95)$ and if this solution is applied to the true model, the expected squared loss would be 3.08, which is much worse than the true optimum.

12.4.3 Identification of Control Factors when the Response Model Contains Alias Terms

Fractional factorial design is a widely used tool to reduce the number of runs in experimental design. The downside of fractional factorial design is that the main effects and higher-order interactions are confounded. For example, in a fractional factorial design with resolution III, the main effects are aliased with the two-factor interaction in the response model. A usual way to address this question is to assume that the interaction is zero and attribute all effects to the main factors. However if the interaction term is important to determine the process/product performance, the loss of this information may be critical. If in the parameter design we cannot differentiate between the effects from the main factors and those from the interaction terms, there is no easy way to identify the optimal levels for the control factors to minimize the variance in the final performance response.

Fractional factorial design usually is used for factor-screening purposes, however if we can use the data from fractional design to make a preliminary assessment of where the optimal levels for control factors may be located, this can help move the design more quickly to the region where the final performance response is most

likely to be optimal and start the full factorial design or other sophisticated designs sooner. So this poses the challenge of how we can solve a parameter design problem if two effects are aliased due to the nature of the data from a fractional factorial design.

Robust optimization provides a useful methodology to address the above challenge if we can include prior information on the alias terms. For example, the prior information can be that both the main factor and interaction term contribute positively to the response value, etc.. To be clear, let us consider the same response model as in (12.18), but assume that, instead of the full factorial design in Table 12.3, only the data from a fractional factorial design is available. The 2^{3-1} design is shown in Table 12.4 where we retain the observations 1, 4, 6, 7 from Table 12.3. At each design point in Table 12.4, replicate 1 is the response value we observed from the design in Table 12.3, in addition, we perform one more run of the experiments and replicate 2 contains the corresponding response value.

We note that the design in Table 12.4 has the defining relation $ABC = I$, so the effects of the main factor A and the interaction BC cannot be differentiated using the data in Table 12.4; similarly the effects of the main factor B and the interaction AC are confounded too. Hence instead of estimating the response model in (12.18), we can only use the data in Table 12.4 to estimate the following model:

$$y = \mu + \boldsymbol{\beta}_1 \tilde{A} + \boldsymbol{\beta}_2 \tilde{B} + \gamma_1 C \,, \tag{12.20}$$

where $\tilde{A} = A + BC$, $\boldsymbol{\beta}_1$ measures the combined effect of the factors A and BC, $\tilde{B} = B + AC$, $\boldsymbol{\beta}_2$ measures the combined effect of the factors B and AC.

Using the same notation as in Sect. 12.4.2, let

$\alpha_1 =$ denote the effect of the main factor A,
$\alpha_2 =$ denote the effect of the main factor B,
$\delta_1 =$ denote the effect of the interaction term AC,
$\delta_2 =$ denote the effect of the interaction term BC.

Given the values for $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, if there is no other information, α_1 and δ_2 can be any values as long as they satisfy $\alpha_1 + \delta_2 = \boldsymbol{\beta}_1$; similarly, α_2 and δ_1 can be any values as long as they satisfy $\alpha_2 + \delta_1 = \boldsymbol{\beta}_2$. However we assume here that quality engineers already know the prior information that: (1) the effects of the main factor A and the interaction BC are in the same direction; and (2) the effects of the main factor B and the interaction AC are in the opposite direction. We can describe the prior information in (1) and (2) in the following constraints:

$$\alpha_1 = \lambda_1 \boldsymbol{\beta}_1, \quad \delta_2 = (1 - \lambda_1)\boldsymbol{\beta}_1, \quad 0 \le \lambda_1 \le 1 \,, \tag{12.21}$$

$$\alpha_2 = \lambda_2 \boldsymbol{\beta}_2, \quad \delta_1 = (1 - \lambda_2)\boldsymbol{\beta}_2, \quad \lambda_2 \ge 1 \,. \tag{12.22}$$

Table 12.4 Comparison of performance responses using canonical and robust optimization approaches (true optimal performance: -19.6)

Dist. of ε	Canonical approach		Robust approach with simultaneous confidence intervals		Robust approach with transformation method	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
$N(0, 0.5)$	-18.7	1.2	-18.2	1.5	-18.4	1.6
$N(0, 1)$	-15.4	6.0	-15.2	3.3	-17.0	3.5
$N(0, 2)$	-9.9	8.7	-10.8	4.9	-15.0	5.4
$N(0, 3)$	-6.3	9.3	-9.0	5.4	-13.2	6.3
$N(0, 4)$	-4.6	9.0	-7.8	5.7	-11.4	6.9

Table 12.5 Comparison of performance responses using canonical, robust, and weighted robust optimization (adapted from [12.13])

ε	Canonical optimization		Robust optimization		Weighted robust opt. with weights (12.10)		Weighted robust opt. with weights (12.11)	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
$N(0, 0.5)$	-18.7	1.2	-18.4	1.6	-18.4	1.4	-18.8	1.0
$N(0, 1)$	-15.4	6.0	-17.0	3.5	-18.0	1.9	-17.8	2.1
$N(0, 2)$	-9.9	8.7	-15.0	5.4	-17.4	2.8	-16.4	3.7
$N(0, 3)$	-6.3	9.3	-13.2	6.3	-17.2	3.0	-15.3	4.8
$N(0, 4)$	-4.6	9.0	-11.4	6.9	-17.0	3.7	-14.7	5.4

We first construct the confidence region B for $(\mu, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \gamma_1)$, the parameters in the response model (12.20). By substituting (12.21) and (12.22) into the optimization problem in (12.19), we have the following equations:

$$\begin{aligned}
&\mathrm{Min}_{\{-1\le A,B\le 1\}}\mathrm{Max}_{(\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\in B}\\
&L(A,B;\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)-g(\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\,,\\
&g(\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\\
&=\mathrm{Min}_{(A,B)}L(A,B;\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\,,\\
&L(A,B;\mu,\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\gamma_1)\\
&=[\mu+\lambda_1\boldsymbol{\beta}_1A+\lambda_2\boldsymbol{\beta}_2B-7.5]^2\\
&\quad+[\gamma_1+(1-\lambda_2)\boldsymbol{\beta}_2A+(1-\lambda_1)\boldsymbol{\beta}_1B]^2+1\,,\\
&0\le\lambda_1\le 1,\ \lambda_2\ge 1\,. \qquad (12.23)
\end{aligned}$$

By solving the optimization problem in (12.23), we will get the solution $(A, B) = (0.17, 0.36)$ with the expected squared loss 1.089. Although this solution seems a little off from the true optimal solution (0.3, 0.3), it still provides valuable information and can guide the design to move quickly to the region closer to the true optimal solution even in the early stage that only the data from the fractional factorial design is available.

We finally comment on the use of the prior information on the main factor effect and the higher-order interaction effect in the formulation of the robust optimization model in (12.23). This information is usually available based on the qualitative knowledge and reasonable judgment of quality engineers. If this information is not available, that is, the values for λ_1 and λ_2 in (12.23) can take any real numbers, we believe robust optimization will not be able to yield a good solution.

References

12.1 G. E. P. Box, W. G. Hunter, J. S. Hunter: *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building* (Wiley, New York 1978)

12.2 G. E. P. Box, N. R. Draper: *Empirical Model-Building and Response Surfaces* (Wiley, New York 1987)

12.3 R. H. Myers, D. C. Montgomery: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments* (Wiley, New York 1995)

12.4 A. I. Khuri, J. A. Cornell: *Response Surfaces: Designs and Analyses* (Marcel Dekker, New York 1996)

12.5 G. E. P. Box, K. B. Wilson: On the experimental attainment of optimum conditions, J. R. Stat. Soc. Ser. B **13**, 1–45 (1951)

12.6 D. Xu, S. L. Albin: Robust optimization of experimentally derived objective functions, IIE Trans. **35**, 793–802 (2003)

12.7 C. F. Wu, M. Hamada: *Experiments: Planning, Analysis, And Parameter Design Optimization* (Wiley, New York 2000)

12.8 E. Erhardt, H. Mai: The search for the optimal paper helicopter, personal communication, (2002) http://www.stat.unm.edu/~erike/projects/Erhardt_-Erik_rsmproj.pdf

12.9 G. E. P. Box, P. Y. T. Liu: Statistics as a catalyst to learning by scientific method, J. Qual. Technol. **31**, 1–15 (1999)

12.10 K. Shimizu, E. Aiyoshi: Necessary conditions for Min-Max problems and algorithms by a relaxation procedure, IEEE Trans. Autom. Control **25**, 62–66 (1980)

12.11 R. G. Miller: *Simultaneous Statistical Inference* (McGraw-Hill, New York 1966)

12.12 R. H. Myers: *Classical and Modern Regression with Applications* (PWS-Kent, Boston 1990)

12.13 D. Xu: Multivariate statistical Modeling and robust optimization in quality engineering, Doctoral Dissertation, Department of Industrial and Systems Engineering, Rutgers University (2001)

12.14 L. Vandenberghe, S. Boyd: Semidefinite programming, SIAM Rev. **38**, 49–95 (1996)

12.15 F. Alizadeh, S. Schmieta: Optimization with semidefinite, quadratic and linear constraints, Rutcor Res. Rep., Rutgers University , 23–97 (1997)

12.16 E. Frazzoli, Z. H. Mao, J. H. Oh, E. Feron: Resolution of conflicts involving many aircraft via semidefinite programming, MIT Research Report , MIT–ICAT 99–5 (1999)

12.17 F. Alizadeh, J. P. Haeberly, M. Nayakkankuppam, M. Overton, S. Schmieta: SDPPACK User's Guide. NYU Computer Science Department Technical Report (1997)

12.18 J. Sturm: Using SeDuMi 1.0x, a Matlab toolbox for optimization over symmetric cones, Optim. Methods Softw. **11**, 625–663 (1999)

12.19 D. Jones, C. Perttunen, B. Stuckman: Lipschitzian optimization without the Lipschitz constant, J. Opt. Theory Appl. **79**, 157–181 (1993)

12.20 M. Bjorkman, K. Holmstrom: Global optimization using the DIRECT algorithm in Matlab, Adv. Model. Opt. **1**, 17–37 (1999)

12.21 G. Taguchi: *System of Experimental Design* (Unipub/Kraus, White Plains 1987)

12.22 W. J. Welch, T. K. Yu, S. M. Kang, J. Sacks: Computer experiments for quality control by robust design, J. Qual. Technol. **22**, 15–22 (1990)

12.23 A. C. Shoemaker, K. L. Tsui, C. F. J. Wu: Economical experimentation methods for robust parameter design, Technometrics **33**, 415–428 (1991)

13. Uniform Design and Its Industrial Applications

Uniform design is a kind of space-filling design whose applications in industrial experiments, reliability testing and computer experiments is a novel endeavor. Uniform design is characterized by uniform scattering of the design points over the experimental domain, and hence is particularly suitable for experiments with an unknown underlying model and for experiments in which the entire experimental domain has to be adequately explored. An advantage of uniform design over traditional designs such as factorial design is that, even when the number of factors or the number of levels of the factors are large, the experiment can still be completed in a relatively small number of runs. In this chapter we shall introduce uniform design, the relevant underlying theories, and the methods of constructing uniform designs in the s-dimensional cube and in the $(q-1)$-dimensional simplex for experiments with mixtures. We shall also give application examples of industrial experiments, accelerated stress testing and computer experiments.

Human history shows that performing experiments systemically is a catalyst to speeding up the process of knowledge discovery. Since the 20th century, when design of experiments was first adopted in agriculture, technology has developed more quickly then ever before. In industry, design of experiments now has an important position in product design and process design. In recent decades, a large amount of theoretical work has been done on design of experiments, and many successful examples of industrial applications are available. For a comprehensive review of the different types of designs, readers may refer to *Ghosh* and *Rao* [13.1]. In this chapter, we shall focus on a type of design called the *uniform design*, whose concept was first introduced in 1978 [13.2] and has now gained popularity and proven to be very successful in industrial applications.

A response in an industrial process may depend on a number of contributing factors. A major objective of an industrial experiment is to explore the relationship between the response and the various causes that may be contributing factors, and to find levels for the contributing factors that optimize the response. Examples of responses are the tensile strength of a material produced from different raw ingredients, the mean time to failure of an electrical component manufactured under different settings of the production equipment, or the yield of a product produced from a chemical process under different reaction conditions. To optimize the response, the relationship between the response and the contributing factors has to be established. If it is difficult to derive the theoretical relationship, experiments may be conducted and statistical methods may be used to establish empirical models or metamodels. When the

form of the model is unknown, one may wish to explore the entire design region by choosing a design whose design points are spread uniformly over the region. Such an objective may be achieved by using uniform design, which was formally introduced in *Fang* [13.3] and *Wang* and *Fang* [13.4]. Figure 2 shows some examples of uniform designs constructed in the two-dimensional square. There are many examples of successful applications of uniform designs in science, engineering and industries. A major multinational automobile manufacturer has recently adopted uniform designs as a standard procedure in product design and process design. A review of applications of uniform designs in chemistry and chemical engineering is given in *Liang* et al. [13.5]. An example of application in quality improvement in electronics manufacturing is given in *Chan* and *Huang* [13.6], *Chan* and *Lo* [13.7] and *Li* et al. [13.8]. Investigations have shown that uniform design performs better at estimating nonlinear problems than other designs, and is robust against model assumptions; see *Zhang* et al. [13.9] and *Xu* et al. [13.10].

Uniform design is different from traditional designs (such as orthogonal arrays and Latin square designs) in that it is not defined in terms of combinatorial structure but rather in terms of the spread of the design points over the entire design region. An advantage of uniform designs over traditional designs is that the former can be used for experiments in which the number of factors and the number of levels of the factor are not small, but a large number of runs is not available. In an experiment with 15 factors and 15 levels on each factor, for example, $225 = 15^2$ runs will be required if an orthogonal array is used, but if a uniform design is used it is possible to complete the experiment in 15 runs. In a Taguchi-type parameter design (*Taguchi* [13.11]), the number of runs required is smaller if uniform designs are used instead of orthogonal arrays. For example, if an $L_{36}(2^3 \times 3^{11})$ orthogonal array is used for the inner and outer arrays, a total of 36×36 runs are required, while if $U_{13}(13^8)$ and $U_{12}(12^{10})$ uniform designs are used instead, $13 \times 12 = 156$ runs will be sufficient [13.12]. Sometimes, to limit the number of runs in an experiment, one may choose designs with a small number of levels, say two- or three-level designs. However, when the behavior of the response is unknown, designs with small numbers of levels are generally unsatisfactory. In Fig. 13.1, all of the two-, three-, four- and five-level designs with equally spaced design points in $[-1, 1]$ (including the points ± 1) wrongly indicate that y decreases as x increases in $[-1, 1]$. Only designs with six or more levels with equally spaced designs points will disclose the peak of y.

A uniform design with n runs, q levels on each of the s factors is denoted by $U_n(q^s)$. Similar notation, for example $U_n(q_1^{s_1} \times q_2^{s_2})$, is used for mixed-level designs. Uniform design tables have been constructed and are available from the website www.math.hkbu.edu.hk/UniformDesign for convenient use. Plots of uniform designs constructed for $n = 2, 5, 8, 20$ are shown in Fig. 13.2. Uniform designs, whose designs points are scattered uniformly over the design region, may be constructed by minimizing a discrepancy. Uniform designs can also be used as space-filling designs in numerical integration. In recent years, many theoretical results on uniform designs have been developed. Readers may refer to *Fang* and *Wang* [13.13], *Fang* and *Hickernell* [13.14], *Hickernell* [13.15], *Fang* and *Mukerjee* [13.16], *Xie* and *Fang* [13.17], *Fang* and *Ma* [13.18, 19], *Fang* et al. [13.20], *Fang* [13.21] and *Hickernell* and *Liu* [13.22].

In what follows, we will use "UD" as an abbreviation for " uniform design". This chapter is organized as follows. Section 13.1 gives a general procedure for conducting an industrial experiment, and gives an example of an application of uniform design in a pharmaceutical experiment which has three contributing factors and where each factor has seven levels. No theoretical model is available for the relationship between these contributing factors and the response (the yield of the process). From the results of the experiment conducted according to a uniform design, several empirical models are proposed, and specific levels for the contributing factors are suggested to maximize the yield. Section 13.2 gives an example of the application of uniform design to accelerated stress testing for determining the

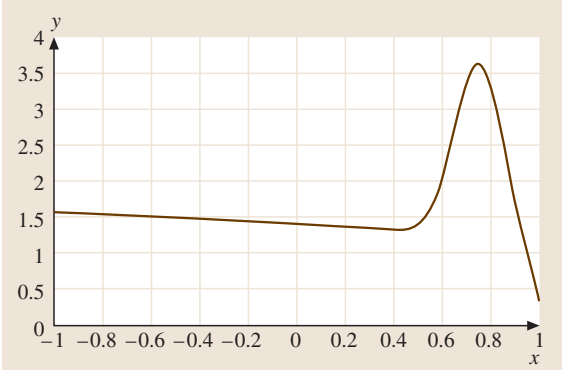


Fig. 13.1 An example of a response curve

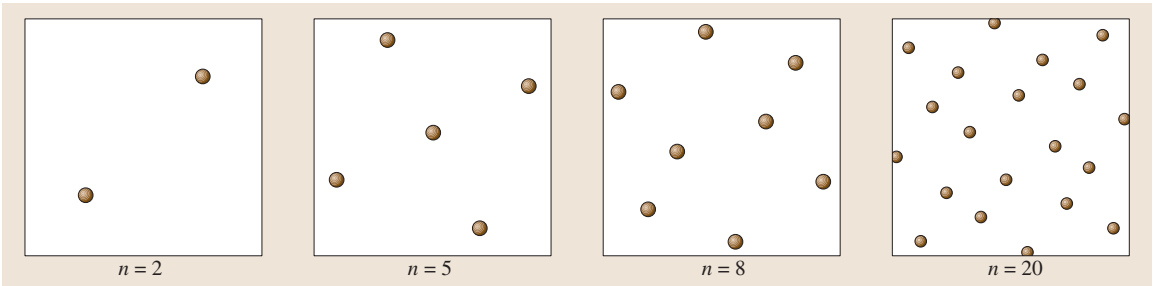


Fig. 13.2 Plots of uniform designs in S^2

median time to failure of an electronics device, with a known theoretical model. The values of the parameters in the theoretical model are determined from the results of accelerated stress testing conducting according to uniform design, and a predicted value for the median time to failure is obtained. Section 13.3 explains when computer experiments can be used for solving practical problems, and illustrates with a simple example on a robot arm how a computer experiment is conducted using uniform design to obtain an approximation of the true theoretical model of the robot arm position. Section 13.4 formally defines uniform design on the s-dimensional cube $[0, 1]^s$ in terms of minimization of the *discrepancy*, and introduces several different discrepancies and their computational formulas. The U-type design, which is used to define a discrete discrepancy, is also introduced. Section 13.5 states that the construction of uniform designs on the s-dimensional cube is an NP-hard problem even when high-power computers are used, and explains how approximate uniform designs can be constructed more easily using U-type designs. Lower bounds for several discrepancies are given, and these lower bounds can be used to indicate how close (in terms of discrepancy) an approximate uniform design is to the theoretical uniform design. Some methods for construction of approximate uniform designs are given. Section 13.6 is devoted to uniform designs for experiments with mixtures in which the contributing factors are proportions of the ingredients in a mixture. It is explained with illustrations how uniform designs can be constructed on the the simplex S^{q-1}, which is the complete design region, and on a subregion of it. Section 13.7 gives the relationships between uniform design and other designs or design criteria, including aberration, orthogonality, supersaturated design, isomorphic design, and equivalent Hadamard matrices. This chapter is concluded briefly in Section 13.8.

13.1 Performing Industrial Experiments with a UD

One purpose of performing industrial experiments is to acquire data to establish quantitative models, if such models cannot be built solely based on theoretical consideration or past experience. Such models can be used to quantify the process, verify a theory or optimize the process. The following steps may be taken as a standard procedure for performing industrial experiments.

1. **Aim.** Specify the aim of the experiment (which may be maximizing the response, defining the operational windows of the contributing and noncontributing factors, etc.), and identify the process response to study.
2. **Factor and domain.** Specify possible contributing factors, and identify the domain of variation of each factor according to experience and practical constraints.
3. **Numbers of levels and runs.** Choose a sufficiently large number of levels for each factor and the total number of runs according to experience, physical consideration and resources available.
4. **Design.** Specify the number of runs and choose a design for the first set of experiment. It is recommended to adopt a UD from the literature or from the website www.math.hkbu.edu.hk/UniformDesign that matches the requirements in Step 3.
5. **Implementation.** Conduct the experiment according to the design chosen in Step 4. Allocate the runs randomly.

6. **Modeling.** Analyze the results using appropriate statistical tools according to the nature of the data. Such tools may include regression methods, ANOVA, Kriging models, neural networks, wavelets, splines, etc. Establish models relating the response to the contributing factors.
7. **Diagnostics.** Make conclusions from the models established in Step 6 to fulfil the aim specified in Step 1.
8. **Further Experiments.** If applicable, perform additional runs of the experiment to verify the results obtained in Steps 6 and 7, or perform subsequent experiments in order to fulfil the aim in Step 1.

The following example illustrates a successful application of UD in an industrial experiment.

Example 13.1: The yield y of an intermediate product in pharmaceutical production depends on the percentages of three materials used: glucose (A), ammonia sulphate (B) and urea (C). The aim of the experiment is to identify the percentages of A, B and C, say x_1, x_2, x_3, which will produce the highest yield. The region for the experiment was defined by the following possible ranges of variation of x_1, x_2, x_3:

$$\text{A: } 8.0 \le x_1 \le 14.0(\%); \quad \text{B: } 2.0 \le x_2 \le 8.0(\%);$$
$$\text{C: } 0.0 \le x_3 \le 0.3(\%). \tag{13.1}$$

It was planned to complete one experiment in not more than eight runs. The levels chosen for the factors are as follows:

x_1: (1)8.0, (2)9.0, (3)10.0, (4)11.0, (5)12.0, (6)13.0, (7)14.0;
x_2: (1)2.0, (2)3.0, (3)4.0, (4)5.0, (5)6.0, (6)7.0, (7)8.0;
x_3: (1)0.00, (2)0.05, (3)0.10, (4)0.15, (5)0.20, (6)0.25, (7)0.30.

Table 13.1 Experiment for the production yield y

No.	$U_7(7^3)$			x_1	x_2	x_3	y
1	1	2	3	8.0	3.0	0.10	7.33
2	2	4	6	9.0	5.0	0.25	5.96
3	3	6	2	10.0	7.0	0.05	6.15
4	4	1	5	11.0	2.0	0.20	9.59
5	5	3	1	12.0	4.0	0.00	8.91
6	6	5	4	13.0	6.0	0.15	6.47
7	7	7	7	14.0	8.0	0.30	4.82

A $U_7(7^3)$ UD was adopted. Table 13.1 shows the $U_7(7^3)$ UD adopted, the layout of the experiment, and the observed response y.

Fitting the data in Table 13.1 with a linear model in x_1, x_2, x_3 gives

$$\hat{y} = 8.1812 + 0.3192x_1 - 0.7780x_2 - 5.1273x_3\,, \tag{13.2}$$

with $R^2 = 0.9444$, $s^2 = 0.3202$, and an F probability of 0.022. The ANOVA is shown in Table 13.2. From (13.2), the maximum value of $\hat{y} = 11.094$ is attained at $x_1 = 14$, $x_2 = 2$ and $x_3 = 0$ within the ranges specified in (13.1).

Fitting the data with a second-degree polynomial by maximizing R^2 gives

$$\hat{y} = 7.0782 + 0.0542x_1^2 - 0.1629x_1x_2 - 0.3914x_1x_3 + 0.1079x_3^2\,, \tag{13.3}$$

with $R^2 = 0.9964$, $s^2 = 0.0309$, and an F probability of 0.007. The ANOVA table is shown in Table 13.3. From (13.3), the maximum value of $\hat{y} = 13.140$ is attained at $(x_1, x_2, x_3) = (14.0, 2.0, 0.0)$, within the ranges specified in (13.1).

On the other hand, fitting the data with a centered second-degree polynomial in the variables $(x_1 - \bar{x}_1)$, $(x_2 - \bar{x}_2)$ and $(x_3 - \bar{x}_3)$ by maximizing R^2 gives

$$\hat{y} = 8.2209 - 0.5652(x_2 - 5) - 4.5966(x_3 - 0.15) - 0.4789(x_1 - 11)^2 + 0.3592(x_1 - 11)(x_2 - 5)\,, \tag{13.4}$$

with $R^2 = 0.9913$, indicating a good fit. The ANOVA table is shown in Table 13.4. From (13.4), the maximum value of $\hat{y} = 11.2212$ is attained at $(x_1, x_2, x_3) = (9.8708, 2.0, 0.0)$, within the ranges specified in (13.1).

The second-degree model (13.2) and the centered second-degree model (13.3) fit the data better than the linear model. The maximum predicted values of $\hat{y}$ given by (13.2–13.4) are between 11.094 and 13.140 when x_1 is between 98.708 and 14, $x_2 = 2$ and $x_3 = 0$ in the design region. These results show that the smaller x_2 and x_3, the larger $\hat{y}$. Zero is the smallest possible value

Table 13.2 ANOVA for a linear model

Source	SS	df	MS	F	P
Regression	16.3341	3	5.4447	17.00	0.022
Error	0.9608	3	0.3203		
Total	17.2949	6			

Table 13.3 ANOVA for a second-degree model

Source	SS	df	MS	F	P
Regression	17.2331	4	4.3083	139.39	0.007
Error	0.0618	2	0.0309		
Total	17.2949	6			

Table 13.4 ANOVA for a centered second-degree model

Source	SS	df	MS	F	P
Regression	17.1445	4	4.2861	56.99	0.0173
Error	0.1504	2	0.0752		
Total	17.2949	6			

of x_3, but x_2 may be extended beyond its smallest value of 2, and x_1 can be extended beyond its largest value of 14 from the boundary of the design region. To explore whether any larger values of maximum y can be achieved outside the design region, further investigation can be carried out by fixing x_3 at 0 and performing two factor experiments with x_1 in the range [8, 16] and x_2 in the range [0, 3].

13.2 Application of UD in Accelerated Stress Testing

Accelerated stress testing is an important method in studying the lifetime of systems. As a result of advancement in technology the lifetime of products is increasing, and as new products emerge quickly product cycle is decreasing. Manufacturers need to determine the lifetime of new products quickly and launch them into the market before another new generation of products emerges. In many cases it is not viable to determine the lifetime of products by testing them under normal operating conditions. To estimate their lifetime under normal operating conditions, accelerated stress testing is commonly used, in which products are tested under high-stress physical conditions. The lifetime of the products are extrapolated from the data obtained using some lifetime models. Many different models, such as the Arrhenius model, the inverse-power-rule model, the proportional-hazards model, etc., have been proposed based on physical or statistical considerations. Readers may refer to *Elsayed* [13.23] for an introduction to accelerated stress testing. In this section we shall give an example to illustrate the application of UD to accelerated stress testing.

Example 13.2: The median time to failure t_0 of an electronics device under the normal operating conditions has to be determined under accelerated stress testing. Theoretical consideration shows that, for such a device, a model of the inverse response type should be appropriate. Under such a model, when the device is operating under voltage V (Volts), temperature T (Kelvin) and relative humidity H (%), its median time to failure t is given by

$$t = a\, V^{-b}\, \mathrm{e}^{c/T}\, \mathrm{e}^{-dH}\,,$$

where a, b, c, d are constants to be determined. Under normal operating conditions, the device operates at $V = 1$, $T = 298$, $H = 60$. The ranges for V, T, H determined for this experiment were 2–5, 353–373, and 85–100, respectively. Logarithmic transformation on the above model gives

$$\ln t = \ln a - b \ln V + c/T - dH\,.$$

An experiment with eight runs and four equally spaced levels on each of $\ln V$, $1/T$ and H was planned. These

Table 13.5 The set up and the results of the accelerated stress test

No.	$U_8(4^3)$			$\ln V$	V	$1/T$	T	H	t
1	1	3	2	0.6931	2	0.0027821	359	90	296.5
2	1	1	3	0.6931	2	0.0026809	373	95	304.3
3	4	1	2	1.6094	5	0.0026809	373	90	95.0
4	4	3	3	1.6094	5	0.0027821	359	95	129.6
5	3	4	1	1.3040	3.68	0.0028328	353	85	278.6
6	3	2	4	1.3040	3.68	0.0027315	366	100	186.0
7	2	4	4	0.9985	2.71	0.0028328	353	100	155.4
8	2	2	1	0.9985	2.71	0.0027315	366	85	234.0

levels were as follows.

$\ln V$: (1)0.6931, (2)0.9985, (3)1.3040, (4)1.6094 ;
$1/T$: (1)0.0026809, (2)0.0027315, (3)0.0027821, (4)0.0028328 ;
H: (1)85, (2)90, (3)95, (4)100 .

The corresponding levels of V and T were

V: (1)2, (2)2.71, (3)3.68, (4)5 ;
T: (1)373, (2)366, (3)359, (4)353 .

The test was performed according to a $U_8(3^4)$ UD. The layout of the experiment and the t values observed are shown in Table 13.5.

Table 13.6 ANOVA for an inverse responsive model

Source	SS	df	MS	F	P
Regression	1.14287	3	0.38096	11.70	0.019
Error	0.13024	4	0.03256		
Total	1.27311	7			

Regression analysis gives

$$\hat{\ln t} = 5.492 - 1.0365 \ln V + 1062/T - 0.02104H ,$$

or

$$\hat{t} = 240.327 V^{-1.0365} \, e^{1062/T - 0.02104H} ,$$

with $R^2 = 0.898$ and $s^2 = 0.0325$. The ANOVA table in Table 13.6 shows a significance level of 0.019. The value of t at the normal operating condition $V = 1$, $T = 298$ and $H = 60$ is estimated to be $\hat{t}_0 = 2400.32$ (hours).

13.3 Application of UDs in Computer Experiments

Indeed, UDs were first used by mathematicians as a space-filling design for numerical integration, and application of UDs in experiments was motivated by the need for effective designs in *computer experiments* in the 1970s [13.2].

The computer can play its role as an artificial means for simulating a physical environment so that experiments can be implemented virtually, if such experiments are not performed physically for some reasons. For example, we do not wish to perform an experiment physically if the experiment may cause casualty. It is not practical to perform a hurricane experiment because we cannot generate and control a hurricane, but if a dynamical model can be established the experiment can be performed virtually on the computer. In such a situation, computer experiments, in which computation or simulation is carried out on the computer, may help study the relation between the contributing factors and the outcome. To perform a computer experiment, levels will have to be set for each of the contributing factors, and in order to have a wide coverage of the entire design region with a limited number of runs, a UD is a good recommendation.

Another use of computer experiments is to establish approximations of known theoretical models if such models are too complicated to handle in practice. From the theoretical model, if computation can be carried out using the computer in evaluating the numerical values of the response y at given values of the variants $x_1, \cdots, x_k$, from the numerical results we can establish metamodels that are good approximations to the theoretical model but yet simple enough for practical use. On the other hand, if the theoretical model is so complicated (for example, represented as a large system of partial differential equations) that it is not even practical to solve it using a computer but if it is possible to observe the values of the response y at different values of $x_1, \cdots, x_k$, we can make use of the computer to establish mathematically tractable empirical models to replace the complicated theoretical model.

In computer experiments, UDs can be used for the selection of representative values of $x_1, \cdots, x_k$ that cover the design region uniformly in a limited number of runs. This is illustrated by an example on water flow in *Fang* and *Lin* [13.24]. Another example of the application of UDs in computer experiments is for real-time control of robotic systems in which the kinematics is described by a system of complicated equations containing various angles, lengths and speeds of movement. Control of robotic systems requires the solution of such a system of equations on a real-time basis at a sufficiently fast speed, which sometimes cannot be achieved because of the intensive computation required (which may involve inversion of high-order Jacobian determinants, etc.). For such a case, computer experiments may be employed, in which the system of equations is solved off-line and the results obtained are used to establish statistical models that are mathematically simple enough to be used for real-time computation. To achieve a sufficiently uniform coverage of the design region, UDs can be used. The fol-

lowing Example 3 is a simplified version of a robot arm in two dimensions which illustrates this application.

Example 13.3: A robot arm on the uv-plane consists of s segments. One end of the first segment is connected to the origin by a rotational join, and the other end of the first segment is connected to one end of the second segment by a rotational join. The other end of the second segment is connected to one end of the third segment a by rotational join, and so on. Let L_j represent the length of the j^{th} segment, θ_1 represent the angle of the first segment with the u-axis, θ_j represent the angle between the $(j-1)^{\text{th}}$ and j^{th} segment, where $0 \le \theta_j \le 2\pi (j = 1, \cdots, s)$. The length between the origin and the end point of the last segment of the robot arm is given by

$$y = f(L_1, \cdots, L_s, \theta_1, \cdots, \theta_s) = \sqrt{u^2 + v^2}\,,$$

$$u = \sum_{j=1}^{s} L_j \cos\Big(\sum_{i=1}^{j} \theta_i\Big)\,,$$

$$v = \sum_{j=1}^{s} L_j \sin\Big(\sum_{i=1}^{j} \theta_i\Big)\,.$$

For simplicity, suppose that $s = 2$. We intend to represent y as a generalized linear function in the variables $L_1, L_2, \theta_1, \theta_2$. A computer experiment is performed with a $U_{28}(28^6)$ UD, in which the values of y were evaluated at different values of L_i and θ_i. The results of the computation is shown in the rightmost column of Table 13.7.

Fitting the data in Table 13.7 with a centered generalized linear regression model with variables $(L_i - 0.5)$, $(\theta_i - \pi)$ and $\cos(\theta_i - \pi)$ $(i = 1, 2, 3)$ using

Table 13.7 Experiment for the robot arm example

No.	$U_{28}(28^6)$						L_1	L_2	l_3	θ_1	θ_2	θ_3	y
1	11	28	6	3	14	20	0.3704	1.0000	0.1852	0.4654	3.0252	4.4215	0.6196
2	17	2	23	27	20	21	0.5926	0.03704	0.8148	6.0505	4.4215	4.6542	0.3048
3	4	14	2	26	9	23	0.1111	0.4815	0.03704	5.8178	1.8617	5.1196	0.4851
4	13	10	4	24	27	13	0.4444	0.3333	0.1111	5.3523	6.0505	2.7925	0.6762
5	14	24	28	8	12	14	0.4815	0.8519	1.0000	1.6290	2.5598	3.0252	0.5636
6	12	5	25	20	4	8	0.4074	0.1482	0.8889	4.4215	0.6982	1.6290	0.7471
7	3	16	10	1	5	10	0.07407	0.5556	0.3333	0.0000	0.9308	2.0944	0.4901
8	21	9	14	7	2	22	0.7407	0.2963	0.4815	1.3963	0.2327	4.8869	1.2757
9	5	11	26	5	18	24	0.1482	0.3704	0.9259	0.9308	3.9561	5.3523	1.0384
10	9	4	12	17	11	27	0.2963	0.1111	0.4074	3.7234	2.3271	6.0505	0.4340
11	15	17	16	10	28	28	0.5185	0.5926	0.5556	2.0944	6.2832	6.2832	1.6667
12	23	19	1	6	19	9	0.8148	0.6667	0.0000	1.1636	4.1888	1.8617	0.7518
13	7	18	20	22	1	18	0.2222	0.6296	0.7037	4.8869	0.0000	3.9561	0.6310
14	18	12	21	4	10	2	0.6296	0.4074	0.7407	0.6981	2.0944	0.2327	0.8954
15	27	8	11	23	13	6	0.9630	0.2593	0.3704	5.1196	2.7925	1.1636	0.4991
16	24	27	17	25	8	12	0.8519	0.9630	0.5926	5.5851	1.6290	2.5598	0.6711
17	20	25	9	21	22	25	0.7037	0.8889	0.2963	4.6542	4.8869	5.5851	1.3362
18	25	6	19	2	24	17	0.8889	0.1852	0.6667	0.2327	5.3523	3.7234	0.3814
19	16	23	5	16	3	4	0.5556	0.8148	0.1482	3.4907	0.4654	0.6982	1.4331
20	8	3	8	9	23	3	0.2593	0.07407	0.2593	1.8617	5.1196	0.4654	0.5408
21	22	15	27	18	26	5	0.7778	0.5185	0.9630	3.9561	5.8178	0.9308	2.1111
22	2	22	13	19	25	16	0.03704	0.7778	0.4444	4.1888	5.5851	3.4907	0.4091
23	26	21	24	13	6	26	0.9259	0.7407	0.8619	2.7925	1.1636	5.8178	2.2386
24	28	13	7	15	17	19	1.0000	0.4444	0.2222	3.2579	3.7234	4.1888	0.6162
25	19	1	3	11	7	15	0.6667	0.0000	0.07407	2.3271	1.3963	3.2579	0.6665
26	6	26	22	12	21	7	0.1852	0.9259	0.7778	2.5598	4.6542	1.3963	1.4167
27	1	7	18	14	15	11	0.0000	0.2222	0.6296	3.0252	3.2579	2.2327	0.5038
28	10	20	15	28	16	1	0.3333	0.8037	0.5185	6.2832	3.4907	0.0000	0.9161

a stepwise procedure in the package SAS gives the model

$$\begin{aligned}\hat{y} = {} & 0.9088 + 0.1760(L_1 - 0.5) + 0.6681(L_2 - 0.5) \\ & + 0.3917(L_3 - 0.5) - 0.2197\cos(\theta_2 - \pi) \\ & - 0.3296\cos(\theta_3 - \pi) \\ & - 0.01919(\theta_1 - \pi)(\theta_2 - \pi) \\ & - 0.5258(L_2 - 0.5)\cos(\theta_3 - \pi) \\ & - 0.0792 L_3 \theta_1 - 0.7622(L_3 - 0.5)\cos(\theta_3 - \pi) \\ & - 0.0114(\theta_1 - \pi)(\theta_3 - \pi) \\ & - 0.0274(\theta_2 - \pi)(\theta_3 - \pi) \\ & + 0.2894\cos(\theta_2 - \pi)\cos(\theta_3 - \pi)\,,\end{aligned}$$

with $R^2 = 0.9868$, $s^2 = 0.0063$, and an F of 0.0000. The ANOVA table is omitted here. Evaluation at different values of L_i and θ_i shows that $\hat{y}$ is a good approximation of y.

13.4 Uniform Designs and Discrepancies

In this section, the formal definition of a UD will be introduced. A UD is, intuitively, a design whose points distribute *uniformly* over the design space. Such uniformity may be achieved by minimizing a discrepancy. There is more than one definition of discrepancy, and different discrepancies may produce different uniform designs.

Without loss of generality, let the design space be the s-dimensional unit cube $C^s = [0, 1]^s$. We represent any point in C^s by $\boldsymbol{x}' = (x_1, \cdots, x_s)$, where $x_1, \cdots, x_s \in [0, 1]$, and the prime denotes the transpose of matrices. For a given positive integer n, a uniform design with n points is a collection of points $\mathcal{P}^* = \{\boldsymbol{x}_1^*, \cdots, \boldsymbol{x}_n^*\} \subset C^s$ such that

$$M(\mathcal{P}^*) = \min M(\mathcal{P})\,,$$

where the minimization is carried out over all $\mathcal{P} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \subset C^s$ with respect to some measure of uniformity, M. One choice for M is the classical L_p-discrepancy adopted in quasi-Monte-Carlo methods [13.25, 26],

$$D_p(\mathcal{P}) = \left(\int_{C^s} \left| \frac{N(\mathcal{P}, [\boldsymbol{0}, \boldsymbol{x}])}{n} - \mathrm{Vol}\,[\boldsymbol{0}, \boldsymbol{x}] \right|^p \mathrm{d}\boldsymbol{x} \right)^{1/p},$$

where $[\boldsymbol{0}, \boldsymbol{x}]$ denotes the interval $[0, x_1] \times \cdots \times [0, x_s]$, $N(\mathcal{P}, [\boldsymbol{0}, \boldsymbol{x}])$ denotes the number of points of $\mathcal{P}$ falling in $[\boldsymbol{0}, \boldsymbol{x}]$, and $\mathrm{Vol}\,[\boldsymbol{0}, \boldsymbol{x}]$ is the volume of the set $[\boldsymbol{0}, \boldsymbol{x}] \in C^s$, which is the distribution function of the uniform distribution on C^s.

The $D_\infty(\mathcal{P})$ discrepancy

$$\max_{\boldsymbol{x} \in C^s} \left| \frac{N(\mathcal{P}, [\boldsymbol{0}, \boldsymbol{x}])}{n} - \mathrm{Vol}\,[\boldsymbol{0}, \boldsymbol{x}] \right|$$

is called the star discrepancy, which is the Kolmogorov–Smirnov statistic used for the goodness-of-fit test.

The L_p-discrepancy is a measure of uniformity of the distribution of points of $\mathcal{P}$ in C^s. The smaller the value of $D_p(\mathcal{P})$, the more uniform the distribution of points of $\mathcal{P}$. The star discrepancy is not as sensitive as the L_p-discrepancy for finite values of p.

The quantity $D_p(\mathcal{P})$ is in general difficult to compute. Let $\boldsymbol{x}_k = (x_{k1}, \cdots, x_{ks})'$ $(k = 1, \cdots, n)$. For $p = 2$, computation can be carried out more efficiently using the following closed-form analytic formula [13.27]:

$$\begin{aligned} D_2(\mathcal{P})^2 = {} & 3^{-s} - \frac{2^{1-s}}{n} \sum_{k=1}^{n} \prod_{l=1}^{s} (1 - x_{kl}^2) \\ & + \frac{1}{x^2} \sum_{k=1}^{n} \sum_{j=1}^{n} \prod_{i=1}^{s} [1 - \max(x_{ki}, x_{ji})]\,. \end{aligned}$$

As pointed out by *Fang* et al. [13.20], the L_2-discrepancy ignores the discrepancy of $\mathcal{P}$ on lower-dimensional subspaces of C^s. To overcome this drawback, *Hickernell* [13.28] proposed the following modified L_2-discrepancy, which includes L_2-discrepancies of projections of $\mathcal{P}$ in all lower dimensional subspaces of C^s

$$\begin{aligned} & D_{2,\,\mathrm{modified}}(\mathcal{P})^2 \\ & = \sum_u \int_{C^u} \left| \frac{N(\mathcal{P}_u, J_{\boldsymbol{x}_u})}{n} - \mathrm{Vol}(J_{\boldsymbol{x}_u}) \right|^p \mathrm{d}\boldsymbol{x}_u\,, \end{aligned} \qquad (13.5)$$

where u is a non-empty subset of the set of coordinate indices $S = \{1, \cdots, s\}$, C^u is the $|u|$-dimensional cube involving the coordinates in u, $|u|$ is the cardinality of u, $\mathcal{P}_u$ is the projection of $\mathcal{P}$ on C^u, $\boldsymbol{x}_u$ is the projection of $\boldsymbol{x}$ on C^u, and $J_{\boldsymbol{x}_u}$ is the projection of a rectangle $J_{\boldsymbol{x}}$ on C^u, which depends on $\boldsymbol{x}$ and is defined based on some specific geometric consideration. Different choices of $J_{\boldsymbol{x}}$ produce discrepancies with different properties, the centered L_2-discrepancy (CD)

(which contains all L_2-discrepancies each calculated using one of the 2^s vertices of C^s as the origin), the wrap-around L_2-discrepancy (WD) (which is calculated after wrapping around each one-dimensional subspace of C^s into a close loop), and others. Closed-form analytic formulas for CD and WD, the most commonly used discrepancies, are displayed below, and corresponding formulas for other discrepancies can be found in *Fang* et al. [13.20] and *Hickernell* [13.28, 29]

$$[\mathrm{CD}(\mathcal{P})]^2 = \left(\frac{13}{12}\right)^s - \frac{2}{n}\sum_{k=1}^{n}\prod_{j=1}^{s}\left(1+\frac{1}{2}|x_{kj}-0.5| - \frac{1}{2}|x_{kj}-0.5|^2\right) + \frac{1}{n^s}\sum_{k=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{s}\left(1+\frac{1}{2}|x_{ki}-0.5| + \frac{1}{2}|x_{ji}-0.5| - \frac{1}{2}|x_{ki}-x_{ji}|\right), \quad (13.6)$$

$$[\mathrm{WD}(\mathcal{P})]^2 = \left(\frac{4}{3}\right)^s + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{s}\left[\frac{3}{2} - |x_{ki}-x_{ji}| \times (1-|x_{ki}-x_{ji}|)\right]. \quad (13.7)$$

The CD is invariant under relabeling of coordinate axes. It is also invariant under reflection of points about any plane passing through the center and parallel to the faces of the unit cube C^s, that is, invariant when the i^{th} coordinate x_i is replaced by $1-x_i$. It follows from the definition that the CD takes into account the uniformity of $\mathcal{P}$ over C^s and also over all projections of $\mathcal{P}$ onto all subspaces of C^s. The uniform designs given in the website www.math.hkbu.edu.hk/UniformDesign are constructed using the CD [13.30].

Another useful discrepancy is called the *discrete discrepancy*, or *categorical discrepancy*. It is defined on the discrete space based on the following U-type designs, and can be used as a vehicle for construction of UDs via U-type designs.

Table 13.8 A design in $\mathcal{U}(6; 3^2\times 2)$

No.	1	2	3
1	1	1	1
2	2	1	2
3	3	2	1
4	1	2	2
5	2	3	1
6	3	3	2

Definition 13.1

A U-type design is an array of n rows and s columns with entries $1, \cdots, q_j$ in the j-th column such that each entry in each column appears the same number of times ($j = 1, \cdots, s$). The collection of all such designs is denoted by $\mathcal{U}(n; q_1\times\cdots\times q_s)$, which is the design space. When all q_j are the same, the design space will be denoted by $\mathcal{U}(n; q^s)$. Designs in $\mathcal{U}(n; q_1\times\cdots\times q_s)$ (where the q_j are distinct) are asymmetric, while designs in $\mathcal{U}(n; q^s)$ are symmetric.

Table 13.8 shows a U-type design in $\mathcal{U}(6; 3^2\times 2)$. Obviously, in a U-type design in $\mathcal{U}(n; q_1\times\cdots\times q_s)$,$n$ must be an integer multiple of q_j for all $j = 1, \cdots, s$.

A discrete discrepancy is defined on $\mathcal{U}(n; q_1\times\cdots\times q_s)$ in terms of two positive numbers $a \neq b$, and is denoted by $D^2(U; a, b)$. The computational formula for $D^2(U; a, b)$ is

$$D^2(U; a, b) = -\prod_{j=1}^{s}\left(\frac{a+(q_j-1)b}{q_j}\right) + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\prod_{j=1}^{s}\widetilde{K}(u_{kj}, u_{lj}), \quad (13.8)$$

where $(u_{k1}, \cdots, u_{ks})$ represents the k-th point in U and

$$\widetilde{K}(u_{kj}, u_{lj}) = \begin{cases} a & \text{if } u_{kj} = u_{lj}, \\ b & \text{if } u_{kj} \neq u_{lj}. \end{cases}$$

13.5 Construction of Uniform Designs in the Cube

In order to construct a uniform design on the continuum $C^s = [0, 1]^s$, we need to search for all possible sets of n points over C^s for a design with minimum discrepancy, which is an NP-hard problem for high-power computers even if n and s are not large. In general, the coordinates of the points in a UD in C^s may be irrational. It can be proved that when $s = 1$, the set $\left\{\frac{1}{2n}, \frac{3}{2n}, \cdots, \frac{2n-1}{2n}\right\}$ with equally spaced points is the n-point uniform design on [0, 1] with CD $= 1/(\sqrt{12}n)$, which is the smallest possible value [13.30]). Since the

design points of a UD distribute *uniformly* over the design region, from the last result on [0, 1] it is natural to expect that values of the coordinates of points in a UD in C^s are either equally spaced or nearly equally spaced on each one-dimensional subspace of C^s. Along this line of thought, while uniform designs defined for the continuum C^s are difficult to find, we can search over the discrete set of U-type designs to construct approximate uniform designs. Computation shows that this approach produces good results. The closeness between the UDs with exactly the minimum discrepancy constructed for C^2 and the approximate UDs constructed from U-type designs for $n = 2, \cdots, 9$ is illustrated in Fig. 13.3 of *Fang* and *Lin* [13.24], and for larger values of n these two types of UDs are practically identical.

Tables of UDs in the website www.math.hkbu.edu.hk/UniformDesign are constructed from U-type designs. Figure 13.2 shows the plots of such designs constructed for $n = 2, 5, 8, 20$ for $s = 2$. An obvious advantage of using U-type designs for construction is that in the UD constructed values of each coordinate of the design are equally spaced. Such designs are much more convenient to use in practice than the exact UDs with irregular values of coordinates constructed for the continuum C^s.

If P is a design consisting of n points $\boldsymbol{x}_1 = (x_{11}, \cdots, x_{1s})', \cdots, \boldsymbol{x}_n = (x_{n1}, \cdots, x_{ns})'$, we shall use the following notations, on different occasions as appropriate, to represent P: $P = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, $P = \{x_{ij}\}_{i=1,\cdots,n;\ j=1,\cdots,s}$, $P = \{x_{ij}\}$, $P = \begin{pmatrix} x_{11} & \cdots & x_{1s} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{ns} \end{pmatrix}$, $P = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{pmatrix}$.

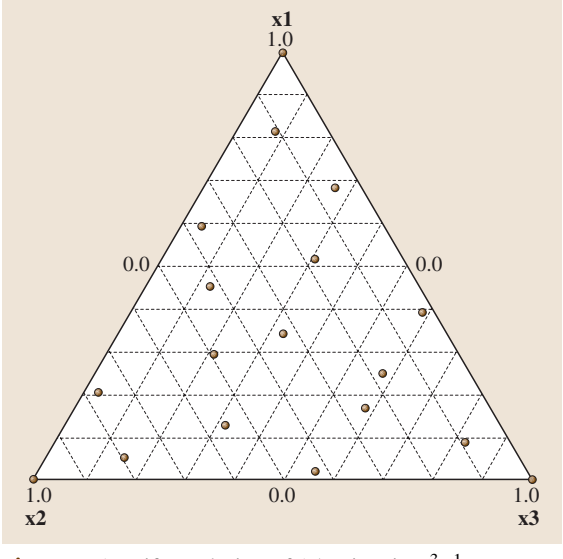


Fig. 13.3 A uniform design of 15 points in S^{3-1}

In the following Definition 13.2, we shall introduce uniform design defined on the discrete set $\mathcal{U}(n; q^s)$.

Definition 13.2

A design $U \in \mathcal{U}(n; q_1 \times \cdots \times q_s)$ is called a uniform design under the measure of discrepancy M if

$$M(U) = \min_{V \in \mathcal{U}(n; q_1 \times \cdots \times q_s)} M(V) .$$

The collection of all such designs is denoted by $U_n(q_1 \times \cdots \times q_s)$. When $q_1 = \cdots = q_s$, U will be called a symmetric design, and $U_n(q_1 \times \cdots \times q_s)$ will be denoted by $U_n(q^s)$.

If $U \in \mathcal{U}(n; q_1 \times \cdots \times q_s)$ is a U-type design consisting of the n points $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n$, where $\boldsymbol{u}_i = (u_{i1}, \cdots, u_{is})'$ $(i = 1, \cdots, n)$, we define $x_{ij} = (u_{ij} - 0.5)/q_j$, so that $\mathcal{P} = \{x_1, \cdots, x_n\} \in C^s$. If M is a discrepancy on C^s, we define $M(U) = M(\mathcal{P})$. Finding UDs in $U_n(q_1 \times \cdots \times q_s)$ by minimizing discrepancies is still an NP-hard problem because of the amount of computation required, even though it is a more manageable task than finding UDs in the continuum C^s. To get around this difficulty, a variety of methods have been proposed by different authors.

For a given discrepancy, and given n and s, it can be seen from the definition that the discrepancy of all designs of n points has a positive lower bound. Thus, lower bounds of discrepancies are used as a benchmark in the construction of UDs or approximate UDs. A UD is a design whose discrepancy equals the lower bound, and a design whose discrepancy is close to the lower bound is a good design.

13.5.1 Lower Bounds of Categorical, Centered and Wrap-Around Discrepancies

(A) Lower Bounds of the Categorical Discrepancy

Let $c(kl)$ be the coincidence number of a pair of elements between rows k and l of a design. Clearly $c(kk) = s$,

and $s-c(kl)$ is the Hamming distance between rows k and l.

Theorem 13.1
A lower bound of the categorical discrepancy in $\mathcal{U}(n; q_1 \times \cdots \times q_s)$ is given by

$$-\prod_{j=1}^{s}\left(\frac{a+(q_j-1)b}{q_j}\right)+\frac{a^s}{n}+\frac{n-1}{n}b^s\left(\frac{a}{b}\right)^{\psi}, \tag{13.9}$$

where $\psi=(\sum_{j=1}^{s} n/q_j - s)/(n-1)$. This lower bound is attained if and only if ψ is an integer and all $c(kl)$ are equal to ψ. When the design space is $\mathcal{U}(n; q^s)$ the above lower bound becomes

$$-\left(\frac{a+(q-1)b}{q}\right)^{s}+\frac{a^s}{n}+\frac{n-1}{n}b^s\left(\frac{a}{b}\right)^{\psi}, \tag{13.10}$$

where $\psi = s(n/q-1)/(n-1)$.

The above lower bonds can be used in searching for UDs in $\mathcal{U}(n; q_1 \times \cdots \times q_s)$.

It is known that block designs have a very good balance structure. Balanced incomplete block (BIB) designs have appeared in many textbooks. *Liu* and *Fang* [13.31] and *Lu* and *Sun* [13.32] found that there is a link between UDs and resolvable BIB designs, a subclass of BIB. Through this link, many UDs can be generated from the large amount of resolvable BIB designs available in the literature. Reader may refer to *Fang* et al. [13.33], *Fang* et al. [13.34] and *Qin* [13.35] for the details.

(B) Lower Bounds of the Wrap-Around L_2-Discrepancy

Values of the wrap-around discrepancy of a designs in $\mathcal{U}(n; q^s)$ can be calculated by (13.7). Let $\alpha_{ij}^k \equiv |x_{ik}-x_{jk}|(1-|x_{ik}-x_{jk}|)$ $(i, j = 1, \cdots, n, i \neq j$ and $k = 1, \cdots, s)$. For any two rows of a design denote the distribution of values of α_{ij}^k by F_{ij}^{α}. *Fang* et al. [13.36] obtained lower bounds for $q = 2, 3$. Recently, *Fang* et al. [13.37] gave lower bounds of the wrap-around discrepancy for any number of levels q as follows:

Theorem 13.2
Lower bounds of the wrap-around L_2-discrepancy on $\mathcal{U}(n; q^s)$ for even and odd q are given by

$$\mathrm{LB}_{\mathrm{even}} = \Delta + \frac{n-1}{n}\left(\frac{3}{2}\right)^{\frac{s(n-q)}{q(n-1)}}\left(\frac{5}{4}\right)^{\frac{sn}{q(n-1)}} \times\left(\frac{3}{2}-\frac{2(2q-2)}{4q^2}\right)^{\frac{2sn}{q(n-1)}}\cdots \times\left(\frac{3}{2}-\frac{(q-2)(q+2)}{4q^2}\right)^{\frac{2sn}{q(n-1)}},$$

$$\mathrm{LB}_{\mathrm{odd}} = \Delta + \frac{n-1}{n}\left(\frac{3}{2}\right)^{\frac{s(n-q)}{q(n-1)}} \times\left(\frac{3}{2}-\frac{2(2q-2)}{4q^2}\right)^{\frac{2sn}{q(n-1)}}\cdots \times\left(\frac{3}{2}-\frac{(q-1)(q+1)}{4q^2}\right)^{\frac{2sn}{q(n-1)}},$$

respectively, where $\Delta = -\left(\frac{4}{3}\right)^s + \frac{1}{n}\left(\frac{3}{2}\right)^s$. A U-type design in $\mathcal{U}(n; q^s)$ is a uniform design under the wrap-around L_2-discrepancy if all its F_{ij}^{α} distributions, for $i \neq j$, are the same. In this case, the WD_2 value of this design achieves the above lower bound.

Fang et al. [13.37] also proposed a powerful algorithm based on Theorem 13.2 and obtained many new UDs.

(C) Lower Bounds of the Centered L_2-Discrepancy

A tight lower bound for the centered L_2-discrepancy is rather difficult to find. *Fang* and *Mukerjee* [13.16] gave a lower bound for the centered L_2-discrepancy on $\mathcal{U}(n; 2^s)$. *Fang* et al. [13.36] gave some improvement of Fang and Mukerjee's results. Recently, *Fang* et al. [13.38] provided a tight lower bound for the centered L_2-discrepancy for $q = 3, 4$. They also proposed an efficient algorithm for searching for UDs.

13.5.2 Some Methods for Construction

The design space $\mathcal{U}(n; q^s)$ contains many poor designs with large values of discrepancy. Confining our search to subspaces in $\mathcal{U}(n; q^s)$ with good designs will significantly reduce the amount of computation. Methods developed along this direction are the *good lattice point method* (see Sect. 1.3 of *Fang* and *Wang* [13.13]), the *Latin square method* and the *extending orthogonal design method* (see *Fang* and *Hickernell* [13.14]). *Ma* and *Fang* [13.39] proposed the *cutting method* that con-

structs a subdesign from a large uniform design. *Fang* and *Qin* [13.40] suggested merging two uniform designs to generate a larger design. Let $U = \{u_{ij}\}$ be a U-type design in $\mathcal{U}(n; q_1 \times \cdots \times q_s)$ and $V = \{v_{kl}\}$ be one in $\mathcal{U}(m; m^t)$. We can construct a new U-type design $D_{U,V}$ by collapsing U and V as follows:

$$D_{U,V} = (\mathbf{1}_m \otimes U \vdots V \otimes \mathbf{1}_n) ,$$

where $\mathbf{1}_n$ is the column vector of ones and $A \otimes B$ is the Kronecker product of $A = (a_{ij})$ and $B = (b_{kl})$ defining by $A \otimes B = (a_{ij} B)$. For example, if

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 1 & 3 \\ 3 & 4 & 2 \\ 4 & 3 & 1 \end{pmatrix} , \quad \text{and} \quad B = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} ,$$

then

$$A \otimes B = \begin{pmatrix} 1 & 2 & 2 & 4 & 4 & 8 \\ 2 & 1 & 4 & 2 & 8 & 4 \\ 2 & 4 & 1 & 2 & 3 & 6 \\ 4 & 2 & 2 & 1 & 6 & 3 \\ 3 & 6 & 4 & 8 & 2 & 4 \\ 6 & 3 & 8 & 4 & 4 & 2 \\ 4 & 8 & 3 & 6 & 1 & 2 \\ 8 & 4 & 6 & 3 & 2 & 1 \end{pmatrix} .$$

If both U and V are uniform designs, *Fang* and *Qin* [13.40] proved that the new design $D_{U,V}$ has the lowest discrepancy in a subclass of $\mathcal{U}(nm; q_1, \times \cdots \times q_s \times m^t)$.

13.6 Construction of UDs for Experiments with Mixtures

Experiments with mixtures are experiments in which the variants are proportions of ingredients in a mixture. An example is an experiment for determining the proportion of ingredients in a polymer mixture that will produce plastics products with the highest tensile strength. Similar experiments are very commonly encountered in industries. A mixture can be represented as $\boldsymbol{x} = (x_1, \cdots, x_q)' \in \{(x_1, \cdots, x_q)' : x_1 + \cdots + x_q = 1; x_1, \cdots, x_q \geq 0\} = S^{q-1}$, where $q \geq 2$ is the number of ingredients in the mixture. The set S^{q-1} is called the $(q-1)$-dimensional simplex. Readers may refer to the monograph by *Cornell* [13.41] and the survey article by *Chan* [13.42] for details of design and modeling in experiments with mixtures. Among the designs for experiments with mixtures, simplex lattice designs have the longest history, followed by simplex centroid designs and axial designs. UDs on S^{q-1}, however, provide a more uniform coverage of the design region than these designs. In this section, we shall explain how UDs on S^{q-1} can be constructed using UDs constructed for C^s.

Suppose that $U = (u_{ki})_{k=1,\cdots,n;i=1,\cdots,q-1}$ is a $U_n(n^{q-1})$ selected from the website. Let $c_{ki} = (u_{ki} - 0.5)/n$ $(k = 1, \cdots, n;\ i = 1, \cdots, q-1)$, and let $\boldsymbol{c}'_k = (c_{k1}, \cdots, c_{k,q-1})$. Then

$$C = \begin{pmatrix} \boldsymbol{c}'_1 \\ \boldsymbol{c}'_2 \\ \vdots \\ \boldsymbol{c}'_n \end{pmatrix}$$

is a UD on $[0, 1]^{q-1}$ from which a UD on S^{q-1} can be constructed. In the construction, special consideration is required because $(x_1, \cdots, x_q)'$ in S^{q-1} is under the constant-sum constraint $x_1 + \cdots + x_q = 1$.

(A) When the Design Region is S^{q-1}

When the design region is the entire simplex S^{q-1}, the variables $x_1, \cdots, x_q$ can take any value in $[0, 1]$ as far as $x_1 + \cdots + x_q = 1$. The following method of constructing UD on S^{q-1} is due to *Wang* and *Fang* [13.43,44] which is also contained in *Fang* and *Wang* [13.13]. For each $\boldsymbol{c}'_k (k = 1, \cdots, n)$ in the above uniform design C, let

$$\begin{aligned} x_{k1} &= 1 - c_{k1}^{1/(q-1)} , \\ x_{k2} &= (1 - c_{k2}^{1/(q-2)}) c_{k1}^{1/(q-1)} , \\ x_{k3} &= (1 - c_{k3}^{1/(q-3)}) c_{k1}^{1/(q-1)} c_{k2}^{1/(q-2)} , \\ &\vdots \\ x_{k,q-1} &= (1 - c_{k,s-1}^{1/1}) c_{k1}^{1/(q-1)} c_{k2}^{1/(q-2)} \cdots c_{k,q-2}^{1/2} , \\ x_{kq} &= c_{k1}^{1/(q-1)} c_{k2}^{1/(q-2)} \cdots c_{k,q-2}^{1/2} c_{k,q-1}^{1/1} . \end{aligned}$$

Let $\boldsymbol{x}'_k = (x_{k1}, \cdots, x_{k,q})$ $(k = 1, \cdots, n)$. Then $\begin{pmatrix} \boldsymbol{x}'_1 \\ \boldsymbol{x}'_2 \\ \vdots \\ \boldsymbol{x}'_n \end{pmatrix}$ is a UD on S^{q-1}. This method of construction is based on the following theory of transformation.

Let $\boldsymbol{x} = (X_1, \cdots, X_s)$ be uniformly distributed on S^{s-1}. Let

$$X_i = C_i^2 \prod_{j=1}^{i-1} S_j^2 (i = 1, \cdots, s-1),$$
$$X_s = \prod_{j=1}^{s-1} S_j^2$$

where

$$S_j = \sin\left(\pi\phi_j/2\right),$$
$$C_j = \cos\left(\pi\phi_j/2\right)$$
$$(j = 1, \cdots, s-1),$$
$$(\phi_1, \cdots, \phi_{s-1}) \in C^{s-1}.$$

Then, we have

(a) $\phi_1, \cdots, \phi_{s-1}$ are mutually independent;
(b) the cumulative distribution function of ϕ_j is

$$F_j(\phi) = \sin^{2(s-j)}(\pi\phi/2),$$
$$(j = 1, \cdots, s-1).$$

With the inverse transformation, the above formulas for $x_{k1}, \cdots, x_{ks}$ follow.

When $q = 3$, this construction is expressed as

$$x_{k1} = 1 - c_{k1}^{1/2},$$
$$x_{k2} = (1 - c_{k2}) c_{k1}^{1/2},$$
$$x_{k3} = c_{k1}^{1/2} c_{k2},$$

and under this transformation a rectangle in S^2 is transformed into a trapezium in S^{3-1}. Figure 13.3 shows a plot of a UD of 15 points on S^{3-1} constructed from the $U_{15}(15^2)$ design

$$\begin{pmatrix} 10 & 15 & 14 & 9 & 6 & 2 & 12 & 13 & 11 & 5 & 1 & 8 & 3 & 4 & 7 \\ 1 & 9 & 3 & 12 & 15 & 13 & 6 & 14 & 17 & 4 & 7 & 5 & 2 & 10 & 8 \end{pmatrix}^{\top}.$$

(B) When There are Restrictions on the Mixture Components

In many cases, lower and upper bounds are imposed on the components in a mixture. For example, in a concrete mixture, the amount of water cannot be less than 10% nor more than 90%. Let $a_i, b_i \in [0, 1]$ $(i = 1, \cdots, q)$, $\boldsymbol{a} = (a_1, \cdots, a_q)'$, $\boldsymbol{b} = (b_1, \cdots, b_q)'$, and let $a = a_1 + \cdots + a_q$ and $b = b_1 + \cdots + b_q$. Define $S_{\boldsymbol{a},\boldsymbol{b}}^{q-1} = \{\boldsymbol{x}' = (x_1, \cdots, x_q) \in S^{q-1} : a_i \le x_i \le b_i (i = 1, \cdots, q)\}$. From $x_1 + \cdots + x_q = 1$ it is not difficult to see that $S_{\boldsymbol{a},\boldsymbol{b}}^{q-1}$ is non-empty if and only if $a \le 1 \le b$, and $S_{\boldsymbol{a},\boldsymbol{b}}^{q-1}$ contains more than one point if and only if $a < 1 < b$. *Fang* and *Yang* [13.45] proposed a method for construction of n-point UDs on $S_{\boldsymbol{a},\boldsymbol{b}}^{q-1}$ using a conditional distribution and the Monte Carlo method. It is more complicated than the method due to *Wang* and *Fang* [13.44], but produces designs with better uniformity. To use this method, the following steps may be followed.

1. Check whether the condition $a < 1 < b$ is satisfied. If this condition is not satisfied, the set $S_{\boldsymbol{a},\boldsymbol{b}}^{q-1}$ is either empty or contains only one point, and in both cases there is no need to construct UDs on $S_{\boldsymbol{a},\boldsymbol{b}}^{q-1}$.
2. Suppose that $a < 1 < b$. Some of the restrictions $a_1 \le x_i \le b_i$ $(i = 1, \cdots, q)$ may be redundant. To remove redundant restrictions, define

$$a_i^0 = \max(a_i, b_i + 1 - b),$$
$$b_i^0 = \min(b_i, a_i + 1 - a)(i = 1, \cdots, q).$$

   The restrictions $a_1^0 \le x_i \le b_i^0$ $(i = 1, \cdots, q)$ do not contains redundant ones, and $a_1 \le x_i \le b_i$ is equivalent to $a_1^0 \le x_i \le b_i^0$ $(i = 1, \cdots, q)$.
3. Reduce the lower bounds to 0 by defining $y_i = \left(x_i - a_i^0\right)/\left[1 - \left(a_1^0 + \cdots + a_q^0\right)\right]$ and $b_i^* = \left(b_i^0 - a_i^0\right)/\left[1 - \left(a_1^0 + \cdots + a_q^0\right)\right](i = 1, \cdots, q)$. Then $a_i^0 \le x_i \le b_i^0$ is equivalent to $0 \le y_i \le b_i^*$ $(i = 1, \cdots, q)$.
4. Define the function $G(c, d, \phi, \Delta, \ell) = \Delta\left(1 - \left[c(1-\phi)^\ell + (1-c)(1-d)^\ell\right]^{1/\ell}\right)$, and follow the steps below to make use the uniform design C on $[0, 1]^{q-1}$ selected above to construct a UD design on the set $S_{\boldsymbol{0},\boldsymbol{b}^*}^{q-1} = \{(y_1, \cdots, y_q)' : 0 \le y_i \le b_i^* (i = 1, \cdots, q)\}$, where $\boldsymbol{b}^* = (b_1^*, \cdots, b_q^*)$. Recall that $\boldsymbol{c}_k' = (c_{k1}, \cdots, c_{k,q-1})$ $(k = 1, \cdots, q-1)$.

Step 1. Let $\Delta_q = 1$,
$d_q = \max\left[0, 1 - (b_1^* + \cdots b_{q-1}^*)/\Delta_q\right]$
$\phi_q = \min\left(1, b_q^*/\Delta_q\right)$.
Let $y_q = G(c_{1,1}, d_q, \phi_q, \Delta_q, q-1)$.

Step 2. Let $\Delta_{q-1} = \Delta_q - y_q$
$d_{q-1} = \max\left[0, 1 - (b_1^* + \cdots b_{q-2}^*)/\Delta_{q-1}\right]$
$\phi_{q-1} = \min\left(1, b_{q-1}^*/\Delta_{q-1}\right)$.
Let $y_{q-1} = G(c_{1,2}, d_{q-1}, \phi_{q-1}, \Delta_{q-1}, q-2)$.
⋮

Step (q−2). Let $\Delta_3 = \Delta_4 - y_4$,
$d_3 = \max\left[0, 1 - \left(b_1^* + b_2^*\right)/\Delta_3\right]$,
$\phi_3 = \min\left(1, b_3^*/\Delta_3\right)$.
Let $y_3 = G(c_{1,q-2}, d_3, \phi_3, \Delta_4, 2)$.

Table 13.9 Construction of UD in $S^{3-1}_{\mathbf{a},\mathbf{b}}$

$\boldsymbol{c}'_1 = (0.625, 0.125)$

(1)°	$\Delta_3 = 1$	$d_3 = 0$	$\phi_3 = 1$	$c_{1,1} = 0.062500$	$y_3 = 0.387628$
(2)°	$\Delta_2 = 0.612372$	$d_2 = 0$	$\phi_2 = 0.816497$	$c_{1,2} = 0.125$	$y_2 = 0.0625$
(3)°	nil	nil	nil	nil	$y_1 = 0.549872$

$(x_1, x_2, x_3) = (0.432577, 0.137500, 0.429923)$.

$\boldsymbol{c}'_2 = (0.125, 0.375)$

(1)°	$\Delta_3 = 1$	$d_3 = 0$	$\phi_3 = 1$	$c_{2,1} = 0.125$	$y_3 = 0.064586$
(2)°	$\Delta_2 = 0.935414$	$d_2 = 0.109129$	$\phi_2 = 0.534522$	$c_{2,2} = 0.375$	$y_2 = 0.25130$
(3)°	nil	nil	nil	nil	$y_1 = 0.684113$

$(x_1, x_2, x_3) = (0.238751, 0.250781, 0.510468)$.

$\boldsymbol{c}'_3 = (0.875, 0.625)$

(1)°	$\Delta_3 = 1$	$d_3 = 0$	$\phi_3 = 1$	$c_{3,1} = 0.875$	$y_3 = 0.646447$
(2)°	$\Delta_2 = 0.853553$	$d_2 = 0$	$\phi_2 = 1$	$c_{3,2} = 0.625$	$y_2 = 0.220971$
(3)°	nil	nil	nil	nil	$y_1 = 0.132583$

$(x_1, x_2, x_3) = (0.587868, 0.232583, 0.179550)$.

$\boldsymbol{c}'_4 = (0.375, 0.875)$

(1)°	$\Delta_3 = 1$	$d_3 = 0$	$\phi_3 = 1$	$c_{4,1} = 0.375$	$y_3 = 0.209431$
(2)°	$\Delta_2 = 0.790569$	$d_2 = 0$	$\phi_2 = 0.632456$	$c_{4,2} = 0.875$	$y_2 = 0.437500$
(3)°	nil	nil	nil	nil	$y_1 = 0.353069$

$(x_1, x_2, x_3) = (0.325659, 0.362500, 0.311841)$.

Step $(q-1)$. Let $\Delta_2 = \Delta_3 - y_3$,
$d_2 = \max(0,\ 1 - b^*_1/\Delta_2)$,

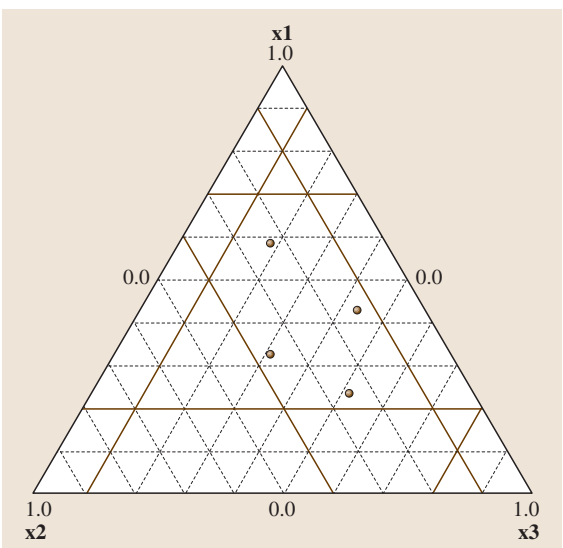


Fig. 13.4 An example of a uniform design with constraints

$\phi_2 = \min\left(1, b^*_2/\Delta_2\right)$.
Let $y_2 = G(c_{1,q-1}, d_2, \phi_2, \Delta_2, 1)$.

Step q. Let $y_1 = 1 - (y_q + \cdots y_2)$.

The point $\boldsymbol{y}'_1 = (y_1, \cdots, y_q)$ is a point for a UD in $S^{q-1}_{\mathbf{0},\boldsymbol{b}^*}$. Let

$$x_1 = \left[1 - \left(a^0_1 + \cdots + a^0_q\right)\right] y_1 + a^0_1 ,$$

$$\vdots$$

$$x_q = \left[1 - \left(a^0_1 + \cdots + a^0_q\right)\right] y_q + a^0_q .$$

The point $\boldsymbol{x}'_1 = (x_1, \cdots, x_q)$ is a point for a UD in $S^{q-1}_{\boldsymbol{a},\boldsymbol{b}}$. Repeat the above with each of $\boldsymbol{c}_2, \cdots, \boldsymbol{c}_{q-1}$ to obtain another $(n-1)$ points $\boldsymbol{y}'_2, \cdots, \boldsymbol{y}'_n$, and thus another $(n-1)$ points $\boldsymbol{x}'_2, \cdots, \boldsymbol{x}'_n$. Let

$$Y = \begin{pmatrix} \boldsymbol{y}'_1 \\ \boldsymbol{y}'_2 \\ \vdots \\ \boldsymbol{y}'_n \} \end{pmatrix} ,$$

and let

$$X = \begin{pmatrix} \boldsymbol{x}_1' \\ \boldsymbol{x}_2' \\ \vdots \\ \boldsymbol{x}_n' \end{pmatrix} .$$

Then Y is a UD on $S^{q-1}_{\boldsymbol{0},\boldsymbol{b}^*}$, and X is a UD on $S^{q-1}_{\boldsymbol{a},\boldsymbol{b}}$.

The following example illustrates construction of a UD of $n = 4$ points on S^{3-1} when there are restrictions on x_1, x_2, x_3.

Example 13.4: Let x_i be subject to the restriction $a_i \le x_i \le b_i$ $(i = 1, 2, 3)$, where $(a_1, a_2, a_3) = (0.2, 0.1, 0.1) = \mathbf{a}'$, $(b_1, b_2, b_3) = (0.7, 0.4, 0.8) = \mathbf{b}'$. Suppose that we want to find a UD with four points on $S^{3-1}_{\boldsymbol{a},\boldsymbol{b}}$. We choose the following $U_4(4^{3-1})$ uniform design U from the website, and from U we construct the following UD, C, on $[0, 1]^{3-1}$ by defining $c_{ki} = (u_{ki} - 0.5)/4$:

$$U = \begin{pmatrix} 3 & 1 \\ 1 & 2 \\ 4 & 3 \\ 2 & 4 \end{pmatrix} , \quad C = \begin{pmatrix} 0.625 & 0.125 \\ 0.125 & 0.375 \\ 0.875 & 0.625 \\ 0.375 & 0.875 \end{pmatrix} .$$

1. We have $a = 0.2 + 0.1 + 0.1 = 0.4$ and $b_1 + b_2 + b_3 = 1.9$. Since the condition $a < 1 < b$ is satisfied, the set $S^{3-1}_{\boldsymbol{a},\boldsymbol{b}}$ contains more than one point and the construction of the UD proceeds.
2. We have

$$\begin{aligned} a_1^0 &= \max(0.2, 0.7 + 1 - 1.9) = 0.2 , \\ b_1^0 &= \min(0.7, 0.2 + 1 - 0.4) = 0.7 , \\ a_2^0 &= \max(0.1, 0.4 + 1 - 1.9) = 0.1 , \\ b_1^0 &= \min(0.4, 0.1 + 1 - 1.9) = 0.4 , \\ a_3^0 &= \max(0.1, 0.8 + 1 - 1.9) = 0.1 , \\ b_1^0 &= \min(0.8, 0.1 + 1 - 0.4) = 0.7 . \end{aligned}$$

3. Define

$$\begin{aligned} y_1 &= (x_1 - 0.2)/0.6 , \\ b_1^* &= (0.7 - 0.2)/0.6 = 5/6 , \\ y_2 &= (x_2 - 0.1)/0.6 , \\ b_2^* &= (0.4 - 0.1)/0.6 = 1/2 , \\ y_3 &= (x_3 - 0.1)/0.6 , \\ b_3^* &= (0.7 - 0.1)/0.6 = 1 . \end{aligned}$$

Then $0.2 \le x_1 \le 0.7$, $0.1 \le x_2 \le 0.4$ and $0.1 \le x_3 \le 0.7$ are equivalent to $0 \le y_1 \le 5/6$, $0 \le y_2 \le 1/2$, $0 \le y_3 \le 1$.

4. Table 13.9 displays the values of Δ_k, d_k, ϕ_k and y_k $(k = 1, 2, 3, 4)$ calculated from the rows c_1', c_2', c_3', c_4' of C.
Hence

$$Y = \begin{pmatrix} 0.387628 & 0.062500 & 0.549872 \\ 0.064586 & 0.251301 & 0.684113 \\ 0.646447 & 0.229071 & 0.132582 \\ 0.209431 & 0.437500 & 0.353069 \end{pmatrix} ,$$

$$X = \begin{pmatrix} 0.432577 & 0.137500 & 0.429923 \\ 0.238751 & 0.250781 & 0.510468 \\ 0.587868 & 0.232583 & 0.179550 \\ 0.325659 & 0.362500 & 0.311841 \end{pmatrix} ,$$

Y is a UD on $S^{3-1}_{\boldsymbol{0},\boldsymbol{b}^*}$, and X is a UD on $S^{q-1}_{\boldsymbol{a},\boldsymbol{b}}$. The plot of the points of X is shown in Fig. 13.4.

13.7 Relationships Between Uniform Design and Other Designs

13.7.1 Uniformity and Aberration

A q^{s-p} factorial design D is uniquely determined by p defining words. A word consists of letters that represent the factors, and the number of letters in a word is called the word-length. The group formed by the p defining words is the defining contrast subgroup of D. Let $A_i(D)$ be the number of words of word-length i in the defining contrast subgroup. If D_1 and D_2 are two regular fractions of a q^{s-p} factorial, and there exists an integer k $(1 \le k \le s)$ such that

$$\begin{aligned} A_1(D_1) = A_1(D_2), \cdots , A_{k-1}(D_1) \\ = A_{k-1}(D_2), A_k(D_1) < A_k(D_2) , \end{aligned}$$

then D_1 is said to have less aberration than D_2. Aberration is a criterion for comparing designs in terms of confounding. The smaller the aberration, the less

confounding the design has, and hence designs with small aberration are preferred. Minimum aberration, as well as maximum resolution, which is also a criterion defined in terms of confounding for comparing designs, are two such commonly used criteria in the literature.

Fang and *Mukerjee* [13.16] proved the following relationship, which connects two seemingly unrelated criteria, CD and aberration, for two-level designs:

$$[\mathrm{CD}(D)]^2 = \left(\frac{13}{12}\right)^s - 2\left(\frac{35}{12}\right)^s + \left(\frac{8}{9}\right)^s \left(1 + \sum_{i=1}^{s} \frac{A_i(D)}{9^i}\right).$$

This relationship shows that minimum CD is essential equivalent to minimum aberration. *Fang* and *Ma* [13.46] extended this result to regular fraction 3^{s-1} designs, and proved the following relationships concerning WD for a regular fractional factorial design q^{s-k} $(q = 2, 3)$:

$$[\mathrm{WD}(D)]^2 = \left(\frac{11}{8}\right)^s - \left(\frac{4}{3}\right)^s + \left(\frac{11}{8}\right) + \sum_{i=1}^{s} \frac{A_i(D)}{11^i} \quad (q = 2),$$

$$[\mathrm{WD}(D)]^2 = -\left(\frac{4}{3}\right)^s + \left(\frac{73}{54}\right)^s \times \left[1 + \sum_{i=1}^{s} \left(\frac{4}{73}\right)^i A_i(D)\right] \quad (q = 3).$$

The last two relationships show that minimum WD and minimum aberration are essentially equivalent.

13.7.2 Uniformity and Orthogonality

An orthogonal array has a balanced structure. In any r columns in an orthogonal array of strength r, combinations of different of $1 \times r$ vectors occur the same number of times. Because of the balanced structure of orthogonal arrays, it is not surprising that an orthogonal array has a small discrepancy and is a uniform design. *Fang* and *Winker* [13.47] showed that many UDs are also orthogonal arrays of strength 2, for example, $U_4(2^3)$, $U_8(2^7)$, $U_{12}(2^{11})$, $U_{12}(2^{11})$, $U_{16}(2^{15})$, $U_9(3^4)$, $U_{12}(3 \times 2^3)$, $U_{16}(4^5)$, $U_{16}(4 \times 2^{12})$, $U_{18}(2 \times 3^7)$ and $U_{25}(25^6)$, and they conjectured that an orthogonal array is a uniform design under a certain discrepancy. *Ma* et al. [13.48] proved this conjecture for complete designs (designs in which all level combinations of the factors appear equally often) and for 2^{s-1} factorials, under L_2-discrepancy.

13.7.3 Uniformity of Supersaturated Designs

A design whose number of runs is equal to the number of effects to estimate is called a *saturated design*. A *supersaturated design* is a design in which the number of runs is less than the number of effects to estimate. In an industrial or scientific experiment, sometimes a large number of possible contributing factors are present, but it is believed that only a few of these factor contribute significantly to the outcome. In this situation of *effect scarcity*, one may use supersaturated designs to identify the major contributing factors. Studies on two- and three-level supersaturated designs are available in the literature [13.49–54].

A supersaturated design can be formed by adding columns to an orthogonal array. Since the number of rows in a supersaturated design is less than the number of columns, a supersaturated design cannot be an orthogonal array. Many criteria have been defined for construction of supersaturated designs that are as close to being orthogonal as possible; they are Ave(s^2), $E(s^2)$, ave(χ^2), and others. *Ma* et al. [13.55] defined a more general criterion, the $D_{\phi,\theta}$ criterion

$$D_{\phi,\theta} = \sum_{1 \le < j \le m} \theta\left(\sum_{u=1}^{q_i} \sum_{v=1}^{q_j} \phi\left|n_{uv}^{(ij)} - \frac{n}{q_i q_j}\right|\right),$$

where $\phi(\cdot)$ and $\theta(\cdot)$ are monotonic increasing functions on $[0, \infty)$, $\phi(0) = \theta(0) = 0$, $n_{uv}^{(ij)}$ is the number of occurrences of the pair (u, v) in the two-column matrix formed by column i and column j of the matrix design. The smaller the value of $D_{\phi,\theta}$, the closer the supersaturated design is to an orthogonal design. Since $n/(q_i q_j)$ is the average number of occurrence of level combinations of the pair (u, v), it is clear that $D_{\phi,\theta} = 0$ for an orthogonal array. *Fang* et al. [13.56] considered a special case of $D_{\phi,\theta}$, denoted by $E(f_{\mathrm{NOD}})$, from which they proposed a way of construction of supersaturated designs. *Fang* et al. [13.56] also proposed a way for constructing supersaturated design with mixed levels. *Fang* et al. [13.57] proposed a way that collapses a uniform design to an orthogonal array for construction of multi-level supersaturated designs. *Fang* et al. [13.33] and *Fang* et al. [13.58] proposed construction of supersaturated designs by a combinatorial approach.

13.7.4 Isomorphic Designs, and Equivalent Hadamard Matrices

Two factorial designs are said to be isomorphic if one can be obtained from the other by exchanging rows and columns and permutating levels of one or more factors. Two isomorphic designs are equivalent in the sense that they produce the same result under the ANOVA model. In the study of factorial designs, a task is to determine whether two designs are isomorphic. To identify two isomorphic designs $d(n, q, s)$ of n runs and s factors each having q levels requires a search over $n!(q!)^s s!$ designs, which is an NP-hard problem even if the values of (n, s, q) are of moderate magnitudes. Some methods have been suggested for reducting the computation load, but such methods are not very satisfactory. The following method using discrepancy suggested by *Ma* et al. [13.59] is a much more efficient alternative.

Given a factorial design $D = d(n, q, s)$ and k $(1 \le k \le s)$, there are $[s!/(k!\,(s-k)!)]$ $d(n, q, s)$ subdesigns. The values of CD of these subdesigns form a distribution $F_k(D)$. It is known that two isomorphic designs $d(n, q, s)$ have the same value of CD and the same distribution $F_k(D)$ for all k, $(1 \le k \le s)$. Based on this, *Ma* et al. [13.59] proposed an algorithm for detecting non-isomorphic designs.

Two Hadamard matrices are said to be equivalent if one can be obtained from the other by some sequence of row and column permutation and negations. To identify whether two Hadamard matrices are equivalent is also an NP-hard problem. A method called the profile method suggested by *Lin* et al. [13.60] can be used, but this method is still not satisfactory. Recently, *Fang* and *Ge* [13.61] proposed a much more efficient algorithm using a symmetric Humming distance and a criterion which has a close relationship with several measures of uniformity. Applying this algorithm, they verified the equivalence of 60 known Hadamard matrices of order 24 and discovered that there are at least 382 pairwise-equivalent Hadamard matrices of order 36.

13.8 Conclusion

In this chapter, we have introduced the *uniform design* (UD) which is a space-filling design characterized by uniform distribution of its design points over the entire experimental domain. Abundant theoretical results on UDs and the relationships between UDs and other well-established design criteria are now available in the literature, as are many successful examples of application of UDs in industry.

Theoretical studies show that UDs are superior, in the sense that establishing uniformity of design by minimizing discrepancies will automatically optimize many other design criteria. An advantage of using UDs in experiments is that, even when the number of factors and the levels of factors are large, the experiment can be conducted in a much smaller number of runs than many other commonly used designs such as factorial designs.

UDs can be used in industrial experiments. Since their design points uniformly cover the design region, UDs are suitable for experiments in which the underlying model is unknown. The UD can be used as a space-filling design for numerical integration and computer experiments, and as a robust design against model specification. For users' convenience, many tables for UDs are documented in the website www.math.hkbu.edu.hk/UniformDesign.

Research in the UD is a new area of study compared with classical areas in experimental designs. Some existing theoretical problems have not yet been solved, and many other problems can be posed. Many successful industrial applications have been recorded, but widespread application of UDs in industries still needs further promotion. We hope that this short chapter can serve as an introduction to the UD, and in the future more researchers and industrial practitioners will join us in studying and applying the UD.

References

13.1 F. Ghosh, C. R. Rao: *Design and Analysis of Experiments*, Handbook of Statistics, Vol. 13 (North Holland, Amsterdam 1996)

13.2 K. T. Fang: Uniform design: application of number-theoretic methods in experimental design, Prob. Stat. Bull. **1**, 56–97 (1978)

13.3 K. T. Fang: The uniform design: application of number-theoretic methods in experimental design, Acta Math. Appl. Sinica **3**, 363–372 (1980)

13.4 Y. Wang, K. T. Fang: A note on uniform distribution and experimental design, KeXue TongBao **26**, 485–489 (1981)

13.5 Y. Z. Liang, K. T. Fang, Q. S. Xu: Uniform design and its applications in chemistry and chemical engineering, Chemomet. Intel. Lab. Syst. **58**, 43–57 (2001)

13.6 L. Y. Chan, G. Q. Huang: Application of uniform design in quality improvement in the manufacture of liquid crystal displays, Proc. 8*th* ISSAT Int. Conf. Reliab. Qual. Des., Anaheim 2002, ed. by H. Pham, M. W. Lu (Int. Soc. Sci. Appl. Technol. (ISSAT), New Brunswick 2002) 245–249

13.7 L. Y. Chan, M. L. Lo: Quality improvement in the manufacture of liquid crystal displays using uniform design, Int. J. Mater. Prod. Technol. **20**, 127–142 (2004)

13.8 R. Li, D. K. J. Lin, Y. Chen: Uniform design: design, analysis and applications, Int. J. Mater. Prod. Technol. **20**, 101–114 (2004)

13.9 L. Zhang, Y. Z. Liang, J. H. Jiang, R. Q. Yu, K. T. Fang: Uniform design applied to nonlinear multivariate calibration by ANN, Anal. Chim. Acta **370**, 65–77 (1998)

13.10 Q. S. Xu, Y. Z. Liang, K. T. Fang: The effects of different experimental designs on parameter estimation in the kinetics of a reversible chemical reaction, Chemomet. Intell. Lab. Syst. **52**, 155–166 (2000)

13.11 G. Taguchi: *Introduction to Quality Engineering* (Asian Production Organization, Tokyo 1986)

13.12 Y. K. Lo, W. J. Zhang, M. X. Han: Applications of the uniform design to quality engineering, J. Chin. Stat. Assoc. **38**, 411–428 (2000)

13.13 K. T. Fang, Y. Wang: *Number-theoretic Methods in Statistics* (Chapman Hall, London 1994)

13.14 K. T. Fang, F. J. Hickernell: *The uniform design and its applications*, Bulletin of The International Statistical Institute, 50th Session, Book 1 (Int. Statistical Inst., Beijing 1995) pp. 339–349

13.15 F. J. Hickernell: Goodness-of-fit statistics, discrepancies and robust designs, Stat. Probab. Lett. **44**, 73–78 (1999)

13.16 K. T. Fang, R. Mukerjee: A connection between uniformity and aberration in regular fractions of two-level factorials, Biometrika **87**, 193–198 (2000)

13.17 M. Y. Xie, K. T. Fang: Admissibility and minimaxity of the uniform design in nonparametric regression model, **83**, 101–111 (2000)

13.18 K. T. Fang, C. X. Ma: The usefulness of uniformity in experimental design. In: *New Trends in Probability and Statistics*, Vol. 5, ed. by T. Kollo, E.-M. Tiit, M. Srivastava (TEV VSP, The Netherlands 2000) pp. 51–59

13.19 K. T. Fang, C. X. Ma: *Orthogonal and Uniform Experimental Designs* (Science Press, Beijing 2001)in Chinese

13.20 K. T. Fang, D. K. J. Lin, P. Winker, Y. Zhang: Uniform design: theory and applications, Technometrics **42**, 237–248 (2000)

13.21 K. T. Fang: Some Applications of Quasi-Monte Carlo Methods in Statistics. In: *Monte Carlo and Quasi-Monte Carlo Methods*, ed. by K. T. Fang, F. J. Hickernell, H. Niederreiter (Springer, Berlin Heidelberg New York 2002) pp. 10–26

13.22 F. J. Hickernell, M. Q. Liu: Uniform designs limit aliasing, Biometrika **89**, 893–904 (2002)

13.23 E. A. Elsayed: *Reliability Engineering* (Addison Wesley, Reading 1996)

13.24 K. T. Fang, D. K. J. Lin: Uniform designs and their application in industry. In: *Handbook on Statistics 22: Statistics in Industry*, ed. by R. Khattree, C. R. Rao (Elsevier, Amsterdam 2003) pp. 131–170

13.25 L. K. Hua, Y. Wang: *Applications of Number Theory to Numerical Analysis* (Springer Science, Beijing 1981)

13.26 H. Niederreiter: *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM CBMS-NSF Regional Conf. Ser. Appl. Math. (SIAM, Philadelphia 1992)

13.27 T. T. Warnock: Computational investigations of low discrepancy point sets. In: *Applications of Number Theory to Numerical Analysis*, ed. by S. K. Zaremba (Academic, New York 1972) pp. 319–343

13.28 F. J. Hickernell: A generalized discrepancy and quadrature error bound, Math. Comp. **67**, 299–322 (1998)

13.29 F. J. Hickernell: Lattice rules: how well do they measure up?. In: *Random and Quasi-Random Point Sets*, ed. by P. Hellekalek, G. Larcher (Springer, Berlin Heidelberg New York 1998) pp. 106–166

13.30 K. T. Fang, C. X. Ma, P. Winker: Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform design, Math. Comp. **71**, 275–296 (2001)

13.31 M. Q. Liu, K. T. Fang: *Some results on resolvable incomplete block designs*, Technical report, MATH-28 (Hong Kong Baptist Univ., Hong Kong 2000) p. 28

13.32 X. Lu, Y. Sun: Supersaturated design with more than two levels, Chin. Ann. Math. B **22**, 183–194 (2001)

13.33 K. T. Fang, G. N. Ge, M. Q. Liu: Construction of optimal supersaturated designs by the packing method, Sci. China **47**, 128–143 (2004)

13.34 K. T. Fang, X. Lu, Y. Tang, J. Yin: Construction of uniform designs by using resolvable packings and coverings, Discrete Math. **274**, 25–40 (2004)

13.35 H. Qin: Construction of uniform designs and usefulness of uniformity in fractional factorial designs. Ph.D. Thesis (Hong Kong Baptist Univ., Hong Kong 2002)

13.36 K. T. Fang, X. Lu, P. Winker: Lower bounds for centered and wrap-around L_2-discrepancies and construction of uniform designs by threshold accepting, J. Complexity **19**, 692–711 (2003)

13.37 K.T. Fang, Y. Tang, J.X. Yin: *Lower bounds for wrap-around L_2-discrepancy and constructions of symmetrical uniform designs*, Technical Report, MATH-372 (Hong Kong Baptist University, Hong Kong 2004)

13.38 K.T. Fang, Y. Tang, P. Winker: Construction of uniform designs via combinatorial optimization, working paper (2004)

13.39 C.X. Ma, K.T. Fang: A new approach to construction of nearly uniform designs, Int. J. Mater. Prod. Technol. **20**, 115–126 (2004)

13.40 K.T. Fang, H. Qin: A note on construction of nearly uniform designs with large number of runs, Stat. Prob. Lett. **61**, 215–224 (2003)

13.41 J.A. Cornell: *Experiments with Mixtures—Designs, Models and the Analysis of Mixture Data* (Wiley, New York 2002)

13.42 L.Y. Chan: Optimal designs for experiments with mixtures: A survey, Commun. Stat. Theory Methods **29**, 2231–2312 (2000)

13.43 Y. Wang, K.T. Fang: Number-theoretical methods in applied statistics (II), Chin. Ann. Math. Ser. B **11**, 384–394 (1990)

13.44 Y. Wang, K.T. Fang: Uniform design of experiments with mixtures, Sci. China Ser. A **39**, 264–275 (1996)

13.45 K.T. Fang, Z.H. Yang: On uniform design of experiments with restricted mixtures and generation of uniform distribution on some domains, Statist. Probab. Lett. **46**, 113–120 (2000)

13.46 K.T. Fang, C.X. Ma: Relationships between uniformity, aberration and correlation in regular fractions 3^{s-1}. In: *Monte Carlo and Quasi-Monte Carlo Methods 2000*, ed. by K.T. Fang, F.J. Hickernell, H. Niederreiter (Springer, Berlin Heidelberg New York 2002) pp. 213–231

13.47 K.T. Fang, P. Winker: *Uniformity and Orthogonality*, Technical Report, MATH-175 (Hong Kong Baptist University, Hong Kong 1998)

13.48 C.X. Ma, K.T. Fang, D.K.J. Lin: A note on uniformity and orthogonality, J. Stat. Plann. Infer. **113**, 323–334 (2003)

13.49 L.Y. Deng, D.K.J. Lin, J.N. Wang: A resolution rank criterion for supersaturated designs, Stat. Sinica **9**, 605–610 (1999)

13.50 D.K.J. Lin: A new class of supersaturated designs, Technometrics **35**, 28–31 (1993)

13.51 D.K.J. Lin: Generating systematic supersaturated designs, Technometrics **37**, 213–225 (1995)

13.52 M.Q. Liu, F.J. Hickernell: $E(s^2)$-optimality and minimum discrepancy in 2-level supersaturated designs, Stat. Sinica **12**(3), 931–939 (2002)

13.53 M.Q. Liu, R.C. Zhang: Construction of $E(s^2)$ optimal supersaturated designs using cyclic BIBDs, J. Stat. Plann. Infer. **91**, 139–150 (2000)

13.54 S. Yamada, D.K.J. Lin: Supersaturated design including an orthogonal base, Cdn. J. Statist. **25**, 203–213 (1997)

13.55 C.X. Ma, K.T. Fang, E. Liski: A new approach in constructing orthogonal and nearly orthogonal arrays, Metrika **50**, 255–268 (2000)

13.56 K.T. Fang, D.K.J. Lin, M.Q. Liu: Optimal mixed-level supersaturated design, Metrika **58**, 279–291 (2003)

13.57 K.T. Fang, D.K.J. Lin, C.X. Ma: On the construction of multi-level supersaturated designs, J. Stat. Plan. Infer. **86**, 239–252 (2000)

13.58 K.T. Fang, G.N. Ge, M.Q. Liu, H. Qin: Construction of uniform designs via super-simple resolvable t-design, Util. Math. **66**, 15–31 (2004)

13.59 C.X. Ma, K.T. Fang, D.K.J. Lin: On isomorphism of fractional factorial designs, J. Complexity **17**, 86–97 (2001)

13.60 C. Lin, W.D. Wallis, L. Zhu: Generalized 4-profiles of Hadamard matrices, J. Comb. Inf. Syst. Sci. **18**, 397–400 (1993)

13.61 K.T. Fang, G.N. Ge: A sensitive algorithm for detecting the inequivalence of Hadamard matrices, Math. Comp. **73**, 843–851 (2004)

14. Cuscore Statistics: Directed Process Monitoring for Early Problem Detection

This chapter presents the background to the Cuscore statistic, the development of the Cuscore chart, and how it can be used as a tool for directed process monitoring. In Sect. 14.1 an illustrative example shows how it is effective at providing an early signal to detect known types of problems, modeled as mathematical signals embedded in observational data. Section 14.2 provides the theoretical development of the Cuscore and shows how it is related to Fisher's score statistic. Sections 14.3, 14.4, and 14.5 then present the details of using Cuscores to monitor for signals in white noise, autocorrelated data, and seasonal processes, respectively. The capability to home in on a particular signal is certainly an important aspect of Cuscore statistics. however, Sect. 14.6 shows how they can be applied much more broadly to include the process model (i. e., a model of the process dynamics and noise) and process adjustments (i. e., feedback control). Two examples from industrial cases show how Cuscores can be devised and used appropriately in more complex monitoring applications. Section 14.7 concludes the chapter with a discussion and description of future work.

The traditional view of statistical process control is that a process should be monitored to detect any aberrant behavior, or what *Deming* [14.1] called "special causes" that are suggested by significant patterns in the data that point to the existence of systematic signals. The timing, nature, size, and other information about the signals can lead to the identification of the signaling factor(s) so that it can (ideally) be permanently eliminated. Conventional Shewhart charts are designed with exactly this philosophy, where the signal they detect is an unexpected spike change in white noise.

Many situations occur, however, where certain process signals are anticipated because they are characteristic of a system or operation. The cumulative score (Cuscore) chart can be devised to be especially sensitive to deviations or signals of an expected type. In general, after working with a particular process, engineers and operators often know – or at least have a belief – about how a process will potentially falter. (Unfortunately, the problem seldom announces its time and location in advance.) For example, consider a process where a valve is used to maintain pressure in a pipeline. Because the valve will experience wear over time, it must be periodically replaced. However, in addition to the usual wear, engineers are concerned that the value may fatigue or fail more rapidly than normal. The Cuscore chart can be used to incorporate this working knowledge and experience into the statistical monitoring function. This concept often has a lot of intuitive appeal for industry practitioners.

After laying the background and theoretical foundation of Cuscores this chapter progresses through signal detection in white noise, autocorrelated data, and seasonal data. Two examples from actual industry settings show how Cuscores can be devised and used appropriately in more complex monitoring applications. The final section of the chapter provides a discussion on how Cuscores can be extended in a framework to include statistical experiments and process control.

14.1 Background and Evolution of the Cuscore in Control Chart Monitoring

Statistical process control (SPC) has developed into a rich collection of tools to monitor a system. The first control chart proposed by *Shewhart* [14.2] is still the most widely used in industrial systems [14.3]. As observational data from the system are plotted on the chart, the process is declared "in control" as long as the points on the chart stay within the control limits. If a point falls outside those limits an "out of control" situation is declared and a search for a special cause is initiated.

Soon practitioners realized that the ability of the Shewhart chart to detect small changes was not as good as its ability to detect big changes. One approach to improve the sensitivity of the chart was to use several additional rules (e.g., *Western Electric rules* [14.4] that signal for a number of consecutive points above the center line, above the warning limits, and so on). Another approach was to design complementary charts that could be used in conjunction with the Shewhart chart but that were better at detecting small changes. *Page* [14.5] and *Barnard* [14.6] developed the cumulative sum (Cusum) chart where past and present data are used in a cumulative way to detect small shifts in the mean. *Roberts* [14.7] and *Hunter* [14.8] proposed the exponentially weighted moving average (EWMA) as another way to detect small changes. This ability comes from the fact that the EWMA statistic can be written as a moving average of the current and past observations, where the weights of the past observations fall off exponentially.

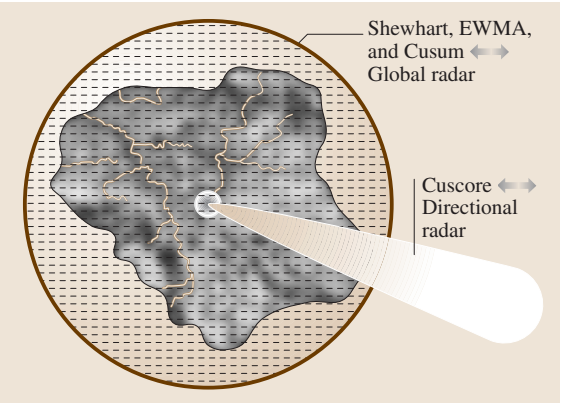


Fig. 14.1 The roles of the Shewhart and Cuscore charts are compared to those of global and directional radar defenses for a small country

Of course the Shewhart, Cusum, and EWMA charts are broadly applicable to many types of process characterizations. Remarkably, the Cuscore chart generalizes the Shewhart, Cusum, and EWMA charts; however, its real benefit is that it can be designed to be a high-powered diagnostic tool for specific types of process characterizations that are not covered by the basic charts. We will develop this result more formally after introducing the Cuscore theory. However, an analogy due to *Box* [14.9] will help to establish the ideas.

Suppose a nation fears aerial attack. As Fig. 14.1 shows, a global radar scanning the full horizon will have a broad coverage of the entire border, but with

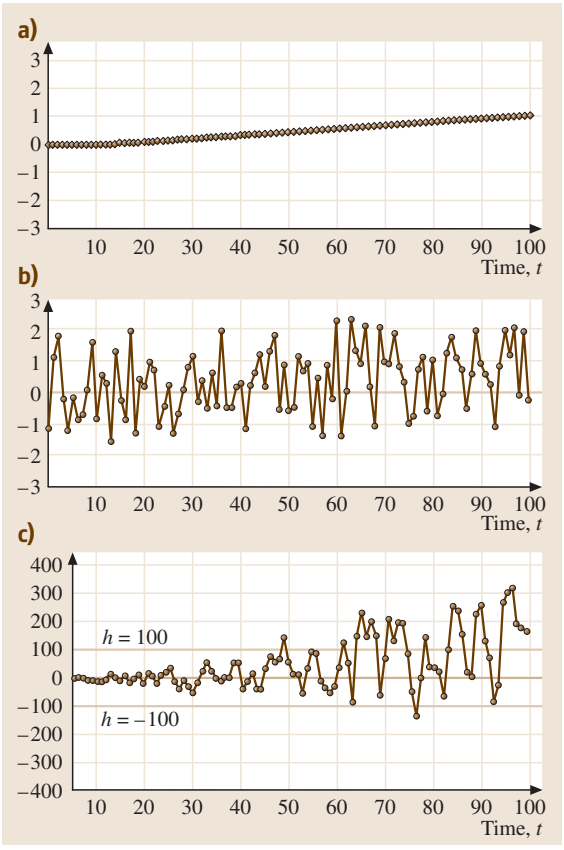


Fig. 14.2a–c Detection of a ramp signal: **(a)** ramp signal beginning at time 10; **(b)** the signal plus white noise consisting of 100 random normal deviates with zero mean and standard deviation $\sigma = 1$; and **(c)** the Cuscore statistic applied to the data of **(b)**

a limited range; this is the analog of the Shewhart, Cusum, and EWMA charts. A directional radar aimed in the direction of likely attack will have a specific zone of the border to cover, but with a long range for early detection; this is the analog of the Cuscore chart.

As a first illustration of the Cuscore chart, let us consider it within the framework of looking for a signal in noise. Suppose we have an industrial process where the objective is to control the output Y_t to a target value T. We may conveniently view the target as the specification and record deviations from the target. Suppose that the process may experience a small drift to a new level over time – a ramp signal. Although corrective actions have been taken to hopefully resolve the process, it is feared that the same problem might reoccur. The components of the process are illustrated in Fig. 14.2 which shows: (a) the ramp signal beginning at time $t = 10$; (b) the signal plus white noise consisting of 100 random normal deviates with zero mean and standard deviation $\sigma = 1$; and (c) the appropriate Cuscore statistic

$$Q_t = \sum_{i=1}^{t} (Y_t - T)t \, .$$

The development of the statistic is also shown in Sect. 14.3. We may note that Fig. 14.2b is equivalent to a Shewhart control chart with upper and lower control limits of $+3\sigma$ and -3σ respectively. The Shewhart chart is relatively insensitive to small shifts in the process, and characteristically, it never detects the ramp signal (Fig. 14.2b). With a decision interval $h = 100$, the Cuscore chart initially detects the signal at time 47 and continues to exceed h at several later time periods (Fig. 14.2c). Although tailored to meet different process monitoring needs, the EWMA and Cusum charts would similarly involve a direct plot of the actual data to look for an unexpected signal in white noise. In this case, since we have some expectation about the signal, i. e., that it is a ramp, we incorporate that information into the Cuscore by multiplying the differences (which are residuals) by t before summing.

Similarly, if demanded by the monitoring needs, the Cuscore can be devised to monitor for a mean shift signal in autocorrelated noise, or for a bump signal in nonstationary noise, or for an exponential signal in correlated noise, or any other combination. Indeed, the Cuscore chart can be designed to look for almost any kind of signal in any kind of noise. The theoretical development of the Cuscore statistic will help illuminate this idea.

14.2 Theoretical Development of the Cuscore Chart

Consider a model of the output of a process determined by adding the process target to an autoregressive integrated moving average (ARIMA) time-series model:

$$Y_t = T + \frac{\theta(B)}{\phi(B)} a_{t0} \, , \qquad (14.1)$$

where B is the backshift operator such that $B^k X_t = X_{t-k}$; $\phi(B)$ and $\theta(B)$ are the autoregressive (AR) and moving average (MA) polynomials parameterized as $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ and $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$ [$(1 - B)\phi(B)$ can be used to difference the process]; the a_t values are independent and identically distributed $N(0, \sigma_a^2)$(i. e., white noise). However, in the model the zero in a_t is added to indicate that the a_{t0} values are just residuals; they are not white-noise innovations unless the model is true. This model is referred to as the null model: the in-control model assuming that no feared signal occurs.

Now assume that an anticipated signal that could appear at some time where γ is some unknown parameter of the signal, and $f(t)$ indicates the nature of the signal:

$$Y_t = T + \frac{\theta(B)}{\phi(B)} a_t + \gamma f(t) \, . \qquad (14.2)$$

This model is referred to as the discrepancy model and is assumed to be true when the correct value for γ is used.

Box and *Ramírez* [14.10, 11] presented a design for the Cuscore chart to monitor for an anticipated signal. It is based on expressing the statistical model in (14.2) in terms of white noise:

$$a_i = a_i(Y_i, X_i, \gamma) \quad \text{for } i = 1, 2, \ldots, t \, , \qquad (14.3)$$

where X_i are the known levels of the input variables. The concept is that we have properly modeled the system so that only white noise remains when the signal is not present. After the data have actually occurred, for each choice of γ, a set of a_i values can be calculated from (14.3). In particular, let γ_0 be some value, possibly different from the true value γ of the parameter. The sequential probability ratio test for γ_0 against some other

value γ_1 has the likelihood ratio

$$LR_t = \prod_{i=1}^{t} \exp\left\{\frac{1}{2\sigma_a^2}\left[a_i^2(\gamma_0) - a_i^2(\gamma_1)\right]\right\}.$$

After taking the log, this likelihood ratio leads to the cumulative sum

$$S_t = \frac{1}{2\sigma_a^2}\sum_{i=1}^{t}\left[a_i^2(\gamma_0) - a_i^2(\gamma_1)\right].$$

Expanding $a_i^2(\gamma)$ around γ_0, letting $\eta = (\gamma_1 - \gamma_0)$, and $d_i = -\frac{\partial a_i}{\partial \gamma}|_{\gamma=\gamma_0}$ we have

$$S_t = \frac{1}{2\sigma_a^2}\sum_{i=1}^{t}\left[2\eta a_i(\gamma_0)d_i(\gamma_0) - \eta^2 d_i^2(\gamma_0)\right]$$

$$= \frac{\eta}{\sigma_a^2}\sum_{i=1}^{t}\left[a_i(\gamma_0)d_i(\gamma_0) - \frac{\eta}{2}d_i^2(\gamma_0)\right].$$

The quantity

$$Q_t = \sum_{i=1}^{t}\left[a_i(\gamma_0)d_i(\gamma_0) - \frac{\eta}{2}d_i^2(\gamma_0)\right] = \sum_{i=1}^{t} q_i \quad (14.4)$$

is referred to as the Cuscore associated with the parameter value $\gamma = \gamma_0$ and d_i is referred to as the detector. The detector measures the instantaneous rate of change in the discrepancy model when the signal appears. *Box* and *Luceño* [14.12] liken its operation to a radio tuner because it is designed in this way to synchronize with any similar component pattern existing in the residuals. Accordingly, it is usually designed to have the same length as the anticipated signal series. The term $\frac{\eta}{2}d_i^2(\gamma_0)$ can be viewed as a reference value around which $a_i(\gamma_0)d_t(\gamma_0)$ is expected to vary if the parameter does not change. The quantity $a_i(\gamma_0)d_t(\gamma_0)$ is equal to Fisher's score statistic [14.13], which is obtained by differentiating the log likelihood with respect to the parameter γ. Thus

$$\frac{\partial}{\partial\gamma}\left[\ln p\left(a_i|\gamma\right)\right]\Big|_{\gamma=\gamma_0} = \frac{\partial}{\partial\gamma}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{t}a_i^2\right)\Big|_{\gamma=\gamma_0}$$

$$= \frac{1}{\sigma^2}\sum_{i=1}^{t}a_i(\gamma_0)d_i(\gamma_0),$$

where $p(a_i|\gamma)$ is the likelihood or joint probability density of a_i for any specific choice of γ and the $a_i(\gamma_0)$ values are obtained by setting $\gamma = \gamma_0$ in (14.3). Since the q_is are in this way a function of Fisher's score function, the test procedure is called the Cuscore. The Cuscore statistic then amounts to looking for a specific signal $f(t)$ that is present when $\gamma \neq \gamma_0$.

To use the Cuscore operationally for process monitoring, we can accumulate q_i only when it is relevant for the decision that the parameter has changed and reset it to zero otherwise. Let Q_t denote the value of the Cuscore procedure plotted at time t, i. e., after observation t has been recorded. Let Q_t^+ and Q_t^- denote the one-sided upper and lower Cuscores respectively as follows:

$$Q_t^+ = \max(0,\, Q_{t-1}^+ + q_t), \quad (14.5a)$$

$$Q_t^- = \min(0,\, Q_{t-1}^- + q_t), \quad (14.5b)$$

where the starting values are $Q_0^+ = Q_0^- = 0$. The one-sided Cuscore is preferable when the system has a long period in the in-control state, during which Q_t would drift and thus reduce the effectiveness of the monitoring chart.

If either Q_t^+ or Q_t^- exceed the decision interval h, the process is said to be out of control. *Box* and *Ramírez* [14.10] showed that an approximation to h can be obtained as a function of the type-I error, α, the magnitude of the change in the parameter $\gamma = (\gamma_1 - \gamma_0)$, and the variance of the as:

$$h = \frac{\sigma_a^2 \ln(1/\alpha)}{\gamma}. \quad (14.6)$$

For simpler models, we could also develop control limits for the Cuscore chart by directly estimating the standard deviation of the Cuscore statistic. For more complex models, control limits may be obtained by using simulation to evaluate the average run length associated with a set of out of control conditions.

14.3 Cuscores to Monitor for Signals in White Noise

Let us now consider the Cuscore statistics for the basic case of monitoring for signals in white noise, which is the assumption underlying the traditional Shewhart, EWMA, and Cusum charts. We will develop them without the reference value in (14.4), but the reference value will help to improve the average run-length performance of the chart when used in practice. We can write the white-noise null model using (14.1) where the ϕ and θ parameters are set equal to zero, i. e.,

$$Y_t = T + a_{t0}.$$

Writing a_{t0} on the left makes it clear that each residual is the difference between the output and the target:

$$a_{t0} = Y_t - T \,. \tag{14.7}$$

If the model is correct and there is no signal, the result will be a white-noise sequence that can be monitored for the appearance of a signal. When the signal does show up, the discrepancy model is thus

$$Y_t = T + a_t + \gamma f(t)$$

which can be equivalently written with the white noise quantity a_t on the left as

$$a_t = Y_t - T - \gamma f(t) \,.$$

The form of the signal will determine the form of the detector and hence the form of the Cuscore.

The Shewhart chart is developed under the assumption of white noise and that the signal for which the chart detects efficiently is a spike signal:

$$f(t) = \begin{cases} 0 & t \neq t_0 \\ 1 & t = t_0 \,. \end{cases} \tag{14.8}$$

For the spike signal in the discrepancy model, the appropriate detector d_t is

$$d_t = -\left.\frac{\partial a_t}{\partial \gamma}\right|_{\gamma=\gamma_0} = 1 \,. \tag{14.9}$$

By (14.4), (14.7), and (14.9) the Cuscore statistic is

$$\begin{aligned} Q_t &= \sum_{i=1}^{t} a_{i0} d_i \\ &= a_{t0} \,, \end{aligned}$$

where the last equality follows since the detector for the spike is only for one period (i.e, the current one) given that the signal series and detector series have the same length. Hence, the Cuscore tells us to plot the current residual, which is precisely the design of the Shewhart chart.

The EWMA chart is developed under the assumption of white noise and that the signal that the chart is designed to detect is an exponential signal with parameter γ:

$$f(t) = \begin{cases} 0 & t > t_0 \\ 1 + \gamma_{t-1} + \gamma_{t-2}^2 + \gamma_{t-3}^3 + \cdots & t \leq t_0 \,. \end{cases}$$

For the exponential signal in the discrepancy model, the appropriate detector d_t is

$$d_t = -\left.\frac{\partial a_t}{\partial \gamma}\right|_{\gamma=\gamma_0} = 1 + \gamma_{t-1} + \gamma_{t-2}^2 + \gamma_{t-3}^3 + \cdots \,. \tag{14.10}$$

By (14.4), (14.7), and (14.10) the appropriate Cuscore statistic is

$$\begin{aligned} Q_t &= \sum_{i=1}^{t} a_{i0} d_i \\ &= a_{t0} + \gamma a_{t0-1} + \gamma^2 a_{t0-2} + \gamma^3 a_{t0-3} + \cdots \,. \end{aligned}$$

Here the Cuscore tells us to sum the current and past residuals, applying an exponentially discounted weight to the past data, which is the design of the EWMA chart.

The Cusum chart is developed under the assumption of white noise and that signal to detect is a step change or mean shift given by

$$f(t) = \begin{cases} 0 & t < t_0 \\ 1 & t \geqslant t_0 \,. \end{cases} \tag{14.11}$$

In this case, the discrepancy model and the detector are the same as for the spike signal. However, since the signal remains in the process, the detector is applied over all periods to give the Cuscore statistic

$$Q_t = \sum_{i=1}^{t} a_{i0} \,.$$

Here the Cuscore tells us to plot the sum of all residuals, which is precisely the design of the Cusum chart.

A variation of the step change is one that lasts only temporarily, which is called a bump signal of length b

$$f(t) = \begin{cases} 1 & t_{0-b+1} \leqslant t \leqslant t_0 \\ 0 & \text{otherwise} \,. \end{cases}$$

When this signal appears in white noise, the detector is applied only as long as the bump, giving the Cuscore statistic

$$Q_t = \sum_{i=1}^{t} a_{i0-b-1} \,.$$

This is equivalent to the arithmetic moving-average (AMA) chart, which is frequently used in financial analysis (e.g., see TraderTalk.com or Investopedia.com).

The ramp signal that may start to appear at time t_{0-r} where r is the duration of the ramp with a final value m is modeled by

$$f(t) = \begin{cases} \frac{m}{r}t & t_{0-r} \leqslant t \leqslant t_0 \\ 0 & \text{otherwise} . \end{cases}$$

The discrepancy model is the same as with the Shewhart chart, but for this signal the detector is given by

$$d_t = -\left.\frac{\partial a_t}{\partial \gamma}\right|_{\gamma=\gamma_0} = t .$$

The Cuscore is hence

$$Q_t = \sum_{i=1}^{t} a_{i0} d_i = \sum_{i=1}^{t} a_{i0} t = \sum_{i=1}^{t} (Y_t - T)\, t$$

as the example in the introduction shows.

14.4 Cuscores to Monitor for Signals in Autocorrelated Data

In many real systems, the assumption of white-noise observations is not even approximately satisfied. Some examples include processes where consecutive measurements are made with short sampling intervals and where quality characteristics are assessed on every unit in order of production. Financial data, such as stock prices and economic indices are certainly not uncorrelated and independent observations. In the case of autocorrelated data the white-noise assumption is violated. Consequently the effectiveness of the Shewhart, Cusum, EWMA, and AMA charts is highly degraded because they give too many false alarms. This point has been made by many authors (e.g., see *Montgomery* [14.14] for a partial list).

Alwan and *Roberts* [14.15] proposed a solution to this problem by modeling the non-random patterns using ARIMA models. They proposed to construct two charts: 1) a common-cause chart to monitor the process, and 2) a special-cause chart on the residuals of the ARIMA model. Extensions of the these charts to handle autocorrelated data have been addressed by several authors. *Vasilopoulos* and *Stamboulis* [14.16] modified the control limits. *Montgomery* and *Mastrangelo* [14.17] and *Mastrangelo* and *Montgomery* [14.18] used the EWMA with a moving center line (MCEWMA). However, when signals occur in autocorrelated data, there is a pattern in the residuals that that the residuals-based control charts do not use. The Cuscore, on the other hand, does incorporate this information through the detector. As we have seen, the detector plays an important role in determining Cuscore statistics but this role is attenuated for autocorrelated data.

As in the previous section, we can use the reference value in practice, but will develop the main result without it. Assuming the null model in (14.1) is invertible, i. e., $|\theta| < 1$, it can be written in terms of the residuals as

$$a_{t0} = (Y_t - T)\frac{\phi(B)}{\theta(B)} . \tag{14.12}$$

The discrepancy model in (14.2) can be equivalently written with the white-noise quantity a_t on the left as

$$a_t = [Y_t - T - \gamma f(t)]\frac{\phi(B)}{\theta(B)} . \tag{14.13}$$

We see that to recover the white-noise sequence in an autocorrelated process, both the residuals and the signal must pass through the inverse filter $\phi(B)/\theta(B)$. Hence, the residuals have time-varying mean $\gamma f(t)\{[\phi(B)/\theta(B)]\}$ and variance σ_a^2. Using (14.13), the detector d_t is

$$d_t = -\left.\frac{\partial a_t}{\partial \gamma}\right|_{\gamma=\gamma_0} = f(t)\frac{\phi(B)}{\theta(B)} . \tag{14.14}$$

By (14.4), (14.13), and (14.14) the Cuscore statistic is

$$\begin{aligned} Q_t &= \sum_{i=1}^{t} a_{i0} d_i \\ &= \sum_{i=1}^{t} \left[(Y_i - T)\frac{\phi(B)}{\theta(B)}\right] f(t)\frac{\phi(B)}{\theta(B)} . \end{aligned}$$

Hu and *Roan* [14.19] mathematically and graphically showed the behavior of the detector for several combinations of signals and time-series models. Their study highlights that the behavior is different for different values of ϕ and θ determined by the stability conditions, the value of the first transient response, and the value of the steady-state response.

As an example, suppose we have the ARMA (1,1) noise model

$$(Y_t - T) - \phi_1(Y_{t-1} - T) = a_{t0} - \theta_1 a_{t0-1}$$

or

$$a_{t0} = (Y_t - T)\frac{1-\phi_1 B}{1-\theta_1 B} \,. \tag{14.15}$$

If the step signal in (14.11) occurs at time t_0, using (14.14) we can determine that a change pattern is produced:

$$d_t = f(t)\frac{\phi(B)}{\theta(B)} = \begin{cases} 0 & t < t_0 \\ 1 & t = t_0 \\ (\theta_1 - \phi_1)\theta_1^{t-(t_0+1)} & t \geqslant t_0 + 1 \,. \end{cases} \tag{14.16}$$

Then the Cuscore statistic is the sum of the product of (14.15) and (14.16).

However, we can see an important issue that arises in autocorrelated data, which is how the time-varying detector is paired with the current residuals. For example, if we assume that we know the time of the step signal or mean shift, there is a match between the residuals and the detector and we use t_0 in the calculation of d_t for the Cuscore. When we do not know the time of the mean shift, there is a mismatch between the residuals and the detector; in this case we make the estimate $\hat{t}_0$ and write the detector as $d_{\hat{t}}$. (When $\hat{t}_0 = t_0$ then $d_{\hat{t}} = d_t$.) The match or mismatch will affect the robustness of the Cuscore chart, as considered for limited cases in *Shu* et al. [14.20] and *Nembhard* and *Changpetch* [14.21]. There is an opportunity to increase the understanding of this behavior through additional studies.

Yet another issue is to determine over how many periods the detector should be used in the case of a finite signal such as a step or a bump. On this point, *Box* and *Luceño* [14.12] use equal lengths for both white-noise and autocorrelated-noise models. Although such an assumption seems intuitive for white-noise models, on open question is whether a longer detector would improve the efficiency of the Cuscore chart in the case of autocorrelated data.

14.5 Cuscores to Monitor for Signals in a Seasonal Process

In this section, we present the first example of a Cuscore application in an industry case. One of the major services of the Red Cross is to manage blood platelet inventory. Platelets are irregularly-shaped colorless bodies that are present in blood. If, for some unexpected reason, sudden blood loss occurs, the blood platelets go into action. Their sticky surface lets them, along with other substances, form clots to stop bleeding [14.22]. *Nembhard* and *Changpetch* [14.21] consider the problem of monitoring blood platelets, where the practical goal is to detect a step shift in the mean of a seasonal process as an indicator that demand has risen or fallen. This information is critical to Red Cross managers, as it indicates a need to request more donors or place orders for more blood with a regional blood bank. A distinction of this problem is that the step shift, although a special cause, is a characteristic of the system. That is, from time to time, shifts in the mean of the process occur due to natural disasters, weather emergencies, holiday travel, and so on. Given the structure of characteristic shifts in this application, directed monitoring is a natural choice.

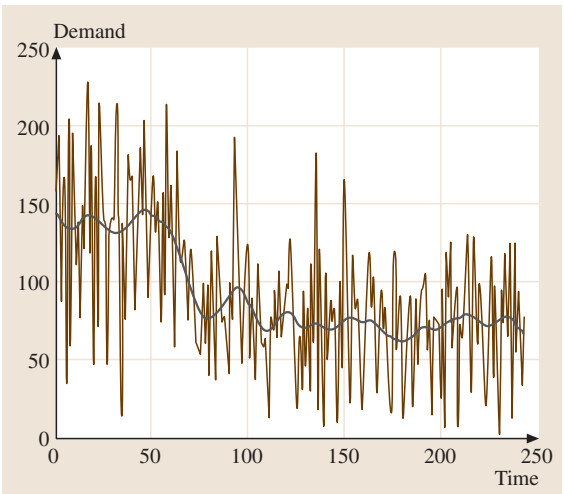


Fig. 14.3 The time-series plot and smooth curve for the quantity of blood platelets ordered from the Red Cross

Figure 14.3 shows the actual time-series data of the demand for platelets from the Red Cross from January 2002 to August 2002 and the smooth curve of the data. The smooth curve suggests that mean of the series has shifted down during the data-collection period. It is easy to visually identify the mean shift in this series. However, it is difficult to conclude that it is a mean shift as it is unfolding. This is the main issue: we want to detect the mean shift as soon as possible in real time. To detect a mean shift in seasonal autocorrelated data, we must use an appropriate time-series model of the original data. Following a three-step model-building

process of model identification, model fitting, and diagnostic checking (*Box*, *Jenkins*, and *Reinsel* [14.23]), we find that an appropriate null model of the data is the ARIMA $(1,0,0)\times(0,1,1)_7$ seasonal model given by

$$\begin{aligned} a_{t0} &= Y_t \frac{\phi(B)}{\theta(B)} = Y_t \frac{(1-B^7)(1+0.281B)}{(1-0.833B^7)} \\ &= Y_t + 0.281Y_{t-1} - Y_{t-7} - 0.281Y_{t-8} \\ &\quad + 0.833a_{t0-7} \,. \end{aligned} \tag{14.17}$$

The discrepancy model is

$$\begin{aligned} a_t &= [Y_t - \gamma f(t)]\frac{\phi(B)}{\theta(B)} \\ &= [Y_t - \gamma f(t)]\frac{(1-B^7)(1+0.281B)}{(1-0.833B^7)} \\ &= Y_t + 0.281Y_{t-1} - Y_{t-7} - 0.281Y_{t-8} - \gamma f(t) \\ &\quad - 0.281\gamma f(t-1) + \gamma f(t-7) + 0.281\gamma f(t-8) \\ &\quad + 0.833a_{t-7} \,. \end{aligned}$$

The detector for the model is

$$\begin{aligned} d_t &= -\left.\frac{\partial a_t}{\partial \gamma}\right|_{\gamma=\gamma_0} \\ &= f(t) + 0.281f(t-1) - f(t-7) \\ &\quad - 0.281f(t-8) + 0.833d_{t-7} \,. \end{aligned} \tag{14.18}$$

Using (14.16) and (14.17) in the one-sided Cuscore statistic of (14.5b) and using a reference value with $\eta = \sigma_a = 31.58$ yields the results shown in Fig. 14.4. The figure also shows that control limits are approximately 196 and −196, which are based on (14.6) with $\alpha = 1/500$. Here the Cuscore chart signals a negative mean shift at observation 67, just two time periods later than the actual occurrence.

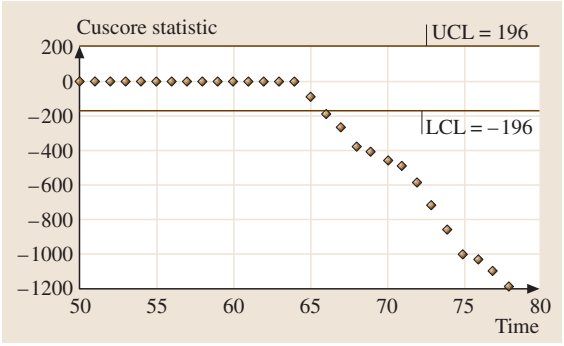


Fig. 14.4 A Cuscore chart for the Red Cross data

This example follows the best-case scenario, which is to predict the time of the occurrence of the mean shift at exactly the time that it really occurs, that is $\hat{t}_0 = t_0$. In such a case, there will be a match between the residuals and the detector, making the use of the Cuscore straightforward. In reality, we are unlikely to have prior information on when the mean shift will occur or, in terms of this application, when there will be a difference in the level of platelets ordered. Consequently, in the determination of the Cuscore statistic there will be a mismatch between the detector and the residuals. The mismatch case is considered fully for this application in *Nembhard* and *Changpetch* [14.21].

14.6 Cuscores in Process Monitoring and Control

As a second example of Cuscore in industry, we now consider a case from *Nembhard* and *Valverde-Ventura* [14.24] where cellular window blinds are produced using a pleating and gluing manufacturing process. Cellular shades form pockets of air that insulate windows from heat and cold. These shades start as 3000-yard rolls of horizontally striped fabric. On the machines, the fabric winds over, under and through several rollers, then a motorized arm whisks a thin layer of glue across it and a pleater curls it into a cell. When the process goes as planned, the crest of the pleat is in the center of the stripe and the finished product is white on the back and has a designer color on the front. When something goes wrong, defects can include a color that bleeds through to the other side, a problem known as "out of registration."

Using a high-speed camera, position data are acquired on the fabric every 20 pleats then a computer compares the edge of the colored band with the target position and measures the deviation (Fig. 14.5). If the two lines match then the deviation is zero and the blind is said to be "in-registration." If the lines do not match, a feedback controller is used to adjust the air cylinder pressure. Unfortunately, as can be seen from the displacement measurements in Fig. 14.5, the feedback controller performed very poorly.

To address this problem, we can use the Box–Jenkins transfer function plus noise and signal model in Fig. 14.6

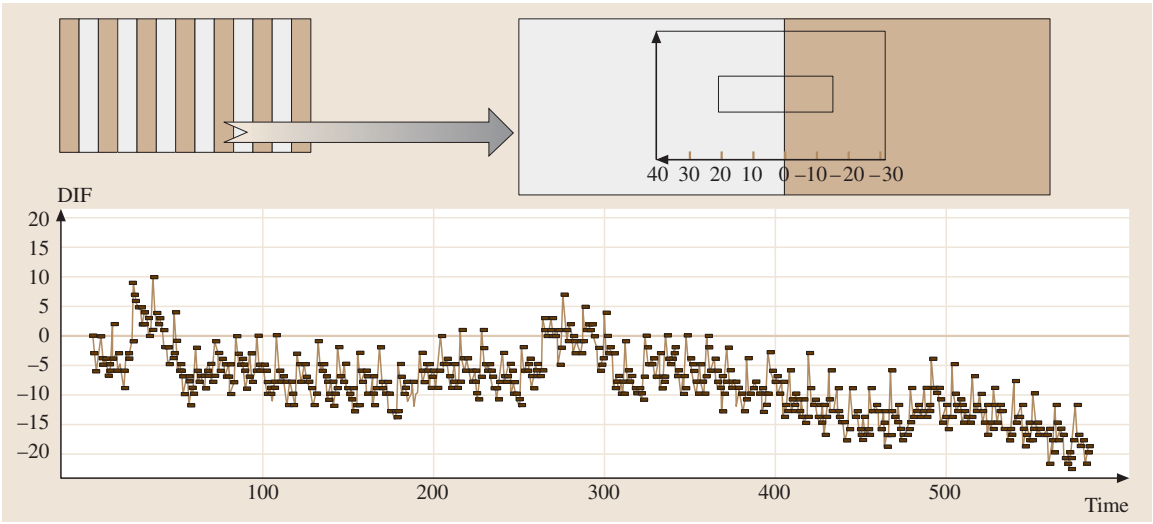


Fig. 14.5 Representation of the measurement of the displacement of the leading edge of the fabric with respect to a fixed point

for process representation. In this model, the output Y_t is the combination of the disturbance term that follows an ARIMA process, as we had in (14.1) and (14.2), plus an input (or explanatory) variable X_t, that is controllable but is affected by the process dynamics S_t. In this case, the combined model of the output in the presence of a signal is:

$$Y_t = \frac{L_2(B)}{L_1(B)} X_{t-k} + a_t \frac{\theta(B)}{\phi(B)} + \gamma \, , \, f(t)$$

where $L_1(B)$ and $L_2(B)$ are the process transfer function polynomials. The control equation tells us how to change X_t over time based on the observed error ε_t. In addition to the process transfer polynomials, the control equation contains the polynomials $L_3(B)$, which describes the noise plus signal Z_t in terms of white noise, and $L_4(B)$, which describes the error ε_t in terms of white noise.

Assuming that minimum variance [or minimum mean-square error (MMSE)] control is applied, we have the null model

$$a_{t0} = \frac{1}{L_4(B)} \varepsilon_t \, . \tag{14.19}$$

The discrepancy model is

$$a_t = \frac{1}{L_4(B)} \varepsilon_t - \gamma f(t) \frac{\phi(B)}{\theta(B)} \, . \tag{14.20}$$

(See *Nembhard* and *Valverde-Ventura* [14.24] for a complete derivation of the null and discrepancy models.)

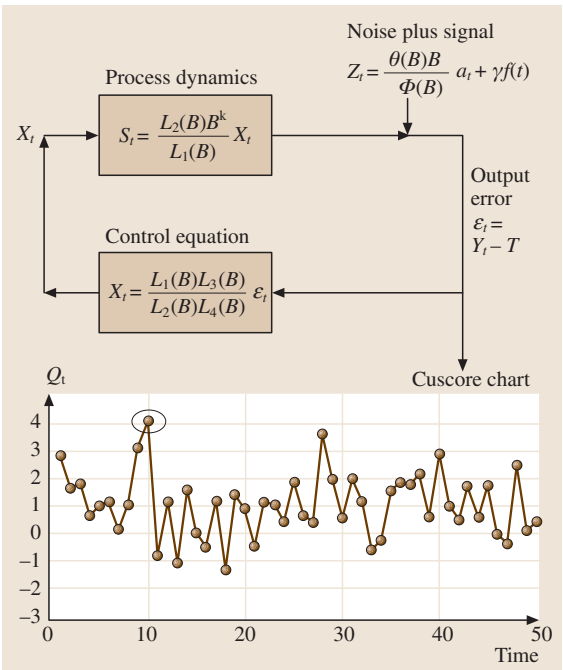


Fig. 14.6 A block diagram showing the input, output, and noise components and the relationship between feedback control and Cuscore monitoring of an anticipated signal

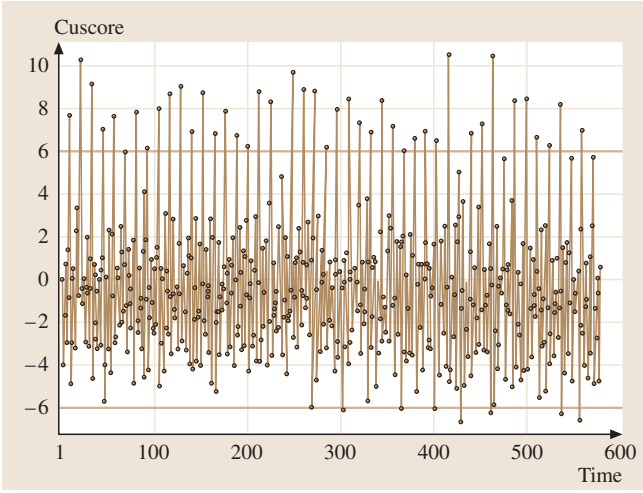


Fig. 14.7 Cuscore chart detects spike signals at every twelfth pleat

Notice that the noise disturbance and signal are assumed to occur after and independently of the process control.

Using (14.20), the detector is

$$d_i = -\left.\frac{\partial a_t}{\partial \gamma}\right|_{\gamma=0} = f(t)\frac{\phi(B)}{\theta(B)} \,. \tag{14.21}$$

Finally, using (14.4), the Cuscore statistic for detecting a signal $f(t)$ hidden in an ARIMA disturbance in an MMSE-controlled process (and omitting the reference value) is given by summing the product of equations (14.19) and (14.21):

$$Q_t = \sum \frac{1}{L_4(B)}\varepsilon_t \frac{\phi(B)}{\theta(B)} f(t) \,. \tag{14.22}$$

For the special case when $k = 1$ (i. e., a responsive system), and the disturbance is white noise, (14.22) simply reduces to the output error, ε_t, which is equivalent to using a Shewhart chart. However, in this pleating and gluing process $k = 2$ and the spike is hidden in an integrated moving-average (IMA) (1, 1) disturbance. The appropriate Cuscore for this case is

$$Q_t = \frac{1}{1+0.84B}\varepsilon_t \,. \tag{14.23}$$

We constructed the Cuscore chart in Fig. 14.7 using (14.23). In this application, during the null operation (i. e., when there is no signal) the Cuscore chart displays observations normally distributed with a mean of zero, and standard deviation $s\sigma_a$. At the moment the spike appears, the corresponding observation belongs to a normal distribution with mean of s^2 and standard deviation of $s\sigma_a$. This mean of s^2 gives the ability for us to observe the spike in the chart.

Note that the Cuscore chart identifies spike signals at pleat numbers 8, 20, 32, etc. In tracking down this problem, it appeared that the printing cylinder used by the supplier to print the fabric was the cause. In that process, the printing consists of passing the fabric over a screen roll with 12 channels. However, one of the twelve stripes had a different width, probably because the printing cylinder was not joined properly at the seam.

14.7 Discussion and Future Work

This chapter focuses on the development and application of Cuscore statistics. Since *Box* and *Ramírez* [14.10,11] presented a design for the Cuscore chart, other work has been done to use them in time series. For example, *Box* and *Luceño* [14.12] suggested monitoring for changes in the parameters of time-series models using Cuscores. *Box* et al. [14.25] and *Ramirez* [14.26] use Cuscores for monitoring industry systems. *Luceño* [14.27] and *Luceño* [14.28] considered average run-length properties for Cuscores with autocorrelated noise. *Shu* et al. [14.20] designed a Cuscore chart that is triggered by a Cusum statistic and uses a generalized likelihood ratio test (GLRT) to estimate the time of occurrence of the signal. These statistical aids help the Cuscore to perform better. *Runger* and *Testik* [14.29] compare the Cuscore and GLRT. *Graves* et al. [14.30] considered a Bayesian approach to incorporating the signal that is in some cases equivalent to the Cuscore. *Harrison* and *Lai* [14.31] develop a sequential probability ratio test (SPRT) that outperforms the Cuscore for the limited cases of data similar to the t-distribution and distributions with inverse polynomial tails.

Although the statistical foundation can be traced back to *Fisher*'s efficient score statistic [14.13], it still needs further development to realize its true potential as a quality engineering tool. Accordingly, *Nembhard* and *Valverde-Ventura* [14.24] developed a framework that may help to guide the development and use of Cuscore statistics in industry applica-

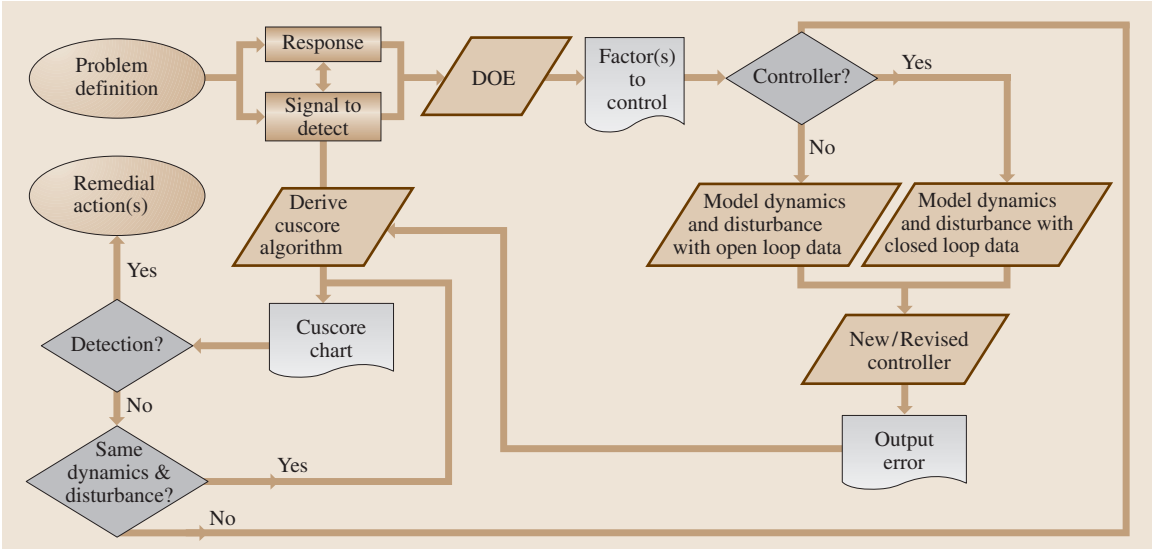


Fig. 14.8 Framework for using Cuscores with DOE and process control

tions, as shown in Fig. 14.8. This framework parallels the define, measure, analyze, improve, and control (DMAIC) approach used in Six Sigma (*Harry* and *Schroeder* [14.32]). The problem-definition step closely parallels the define step in DMAIC. Design of experiments (DOE) helps us to measure and analyze the process, the second two DMAIC steps. From DOE we develop an understanding of the factors to control, so we can then adjust and monitor in keeping with the last two DMAIC steps. The monitoring in this case is accomplished using a Cuscore chart.

The Cuscore is a natural fit with the DMAIC approach as it strives to incorporate what we learn about the problem into the solution. Some consideration needs to be given to the system to establish a clear understanding of the response, the expected signal to be detected, and the relationship between the two. More specifically, for the Cuscore to be applicable we should be able to describe how the signal might modify the response and, therefore, the output error.

This framework also recognizes that in many industrial systems, using only SPC to monitor a process will not be sufficient to achieve acceptable output. Real processes tend to drift away from target, use input material from different suppliers, and are run by operators who may use different techniques. For these and many other reasons, a system of active adjustment using engineering process control (EPC) is often necessary. *Box* and *Jenkins* [14.33] pioneered the integration of SPC and EPC to monitor and adjust industrial processes jointly by demonstrating the interrelationships between adaptive optimization, adaptive quality control, and prediction. *Box* and *Kramer* [14.34] revived the discussion on the complementary roles of SPC and EPC. Since then, many other authors have addressed the joint monitoring and adjustment of industrial processes. *Montgomery* and *Woodall* [14.35] give over 170 references in a discussion paper on statistically based process monitoring and control. Others since include *Shao* [14.36]; *Nembhard* [14.37]; *Nembhard* and *Mastrangelo* [14.38]; *Tsung*, *Shi*, and *Wu* [14.39]; *Tsung* and *Shi* [14.40]; *Ruhhal*, *Runger*, and *Dumitrescu* [14.41]; *Woodall* [14.42]; *Nembhard* [14.43]; *Nembhard*, *Mastrangelo*, and *Kao* [14.44]; and *Nembhard* and *Valverde–Ventura* [14.45]. The texts by *Box* and *Luceño* [14.12] and *del Castillo* [14.46] also address the topic.

In addition to those issues addressed in Sect. 14.5 for autocorrelated data, future work that will further advance the area of Cuscore statistics include their integration with suboptimal controllers, which are often used in practice. There is also a great need to expand the understanding of the robustness of Cuscores to detect signals (other than the one specifically designed for), to develop ways to detect multiple signals then identify or classify them once an out-of-control condition occurs, and to develop multivariate Cuscore detection capabilities.

References

14.1 W. E. Deming: *Out of the Crisis* (Center for Advanced Engineering Studies, Cambridge 1986)

14.2 W. A. Shewhart: Quality control charts, Bell Sys. Tech. J. **5**, 593–603 (1926)

14.3 Z. G. Stoumbos, M. R. Reynolds Jr., T. P. Ryan, W. H. Woodall: The state of statistical process control as we proceed into the 21st century, J. Am. Stat. Assoc. **451**, 992–998 (2000)

14.4 Western Electric: *Statistical Quality Control Handbook* (Western Electric Corp., Indianapolis 1956)

14.5 E. S. Page: Continuous inspection schemes, Biometrika **41**, 100–114 (1954)

14.6 G. A. Barnard: Control charts and stochastic processes, J. R. Stat. Soc. B **21**, 239–271 (1959)

14.7 S. W. Roberts: Control chart tests based on geometric moving averages, Technometrics **1**, 239–250 (1959)

14.8 J. S. Hunter: The exponentially weighted moving average, J. Qual. Technol. **18**, 203–210 (1986)

14.9 G. E. P. Box: Sampling and Bayes' inference in scientific modeling and robustness, J. R. Stat. Soc. A **143**, 383–430 (1980)

14.10 G. E. P. Box, J. Ramírez: *Sequential Methods in Statistical Process Monitoring: Sequential Monitoring of Models*, CQPI Report No. 67 (Univ. Wisconsin, Madison 1991)

14.11 G. E. P. Box, J. Ramírez: Cumulative score charts, Qual. Reliab. Eng. Int. **8**, 17–27 (1992). Also published as *Report No. 58* (Univ. Wisconsin, Madison 1992)

14.12 G. E. P. Box, A. Luceño: *Statistical Control by Monitoring and Feedback Adjustment* (Wiley, New York 1997)

14.13 R. A. Fisher: Theory of statistical estimation, Proc. Cambridge Philos. Soc. **22**, 700–725 (1925)

14.14 D. C. Montgomery: *Introduction to Statistical Process Control*, 5th edn. (Wiley, New York 2005)

14.15 L. Alwan, H. V. Roberts: Time-series modeling for statistical process control, J. Bus. Econ. Stat. **6**, 87–95 (1988)

14.16 A. V. Vasilopoulos, A. P. Stamboulis: Modification of control chart limits in the presence of correlation, J. Qual. Technol. **10**, 20–30 (1978)

14.17 D. C. Montgomery, C. M. Mastrangelo: Some statistical process control methods for autocorrelated data, J. Qual. Technol. **23**, 179–204 (1991)

14.18 C. M. Mastrangelo, D. C. Montgomery: SPC with correlated observations for the chemical and process industries, Qual. Reliab. Eng. Int. **11**, 79–89 (1995)

14.19 S. J. Hu, C. Roan: Change patterns of time series-based control charts, J. Qual. Technol. **28**, 302–312 (1996)

14.20 L. Shu, D. W. Apley, F. Tsung: Autocorrelated process monitoring using triggered cuscore charts, Qual. Reliab. Eng. Int. **18**, 411–421 (2002)

14.21 H. B. Nembhard, P. Changpetch: Directed monitoring of seasonal processes using cuscore statistics, Qual. Reliab. Eng. Int. , to appear (2006)

14.22 Franklin Institute Online: *Blood Platelets* (http://www.fi.edu/biosci/blood/platelet.html, 2004)

14.23 G. E. P. Box, G. M. Jenkins, G. C. Reinsel: *Time Series Analysis, Forecasting And Control*, 3rd edn. (Prentice Hall, Englewood Cliffs 1994)

14.24 H. B. Nembhard, R. Valverde-Ventura: A framework for integrating experimental design and statistical control for quality improvement in manufacturing, J. Qual. Technol. **35**, 406–423 (2003)

14.25 G. Box, S. Graves, S. Bisgaard, J. Van Gilder, K. Marko, J. James, M. Seifer, M. Poublon, F. Fodale: *Detecting Malfunctions In Dynamic Systems*, Report No. 173 (Univ. Wisconsin, Madison 1999)

14.26 J. Ramírez: Monitoring clean room air using cuscore charts, Qual. Reliab. Eng. Int. **14**, 281–289 (1992)

14.27 A. Luceño: Average run lengths and run length probability distributions for cuscore charts to control normal mean, Comput. Stat. Data Anal. **32**, 177–195 (1999)

14.28 A. Luceño: Cuscore charts to detect level shifts in autocorrelated noise, Qual. Technol. Quant. Manag. **1**, 27–45 (2004)

14.29 G. Runger, M. C. Testik: Control charts for monitoring fault signatures: Cuscore versus GLR, Qual. Reliab. Eng. Int. **19**, 387–396 (2003)

14.30 S. Graves, S. Bisgaard, M. Kulahci: A Bayes-adjusted cumulative sum. Working paper (2002)

14.31 P. J. Harrison, I. C. H. Lai: Statistical process control and model monitoring, J. Appl. Stat. **26**, 273–292 (1999)

14.32 M. Harry, R. Schroeder: *Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations* (Random House, New York 2000)

14.33 G. E. P. Box, G. M. Jenkins: Some statistical aspects of adaptive optimization and control, J. R. Stat. Soc. B **24**, 297–343 (1962)

14.34 G. E. P. Box, T. Kramer: Statistical process monitoring and feedback adjustment—a discussion, Technometrics **34**, 251–285 (1992)

14.35 D. C. Montgomery, W. H. Woodall (Eds.): A discussion of statistically-based process monitoring and control, J. Qual. Technol. **29**, 2 (1997)

14.36 Y. E. Shao: Integrated application of the cumulative score control chart and engineering process control, Stat. Sinica **8**, 239–252 (1998)

14.37 H. B. Nembhard: Simulation using the state-space representation of noisy dynamic systems to determine effective integrated process control designs, IIE Trans. **30**, 247–256 (1998)

14.38 H. B. Nembhard, C. M. Mastrangelo: Integrated process control for startup operations, J. Qual. Technol. **30**, 201–211 (1998)
14.39 F. Tsung, J. Shi, C. F. J. Wu: Joint monitoring of PID-controlled processes, J. Qual. Technol. **31**, 275–285 (1999)
14.40 F. Tsung, J. Shi: Integrated design of run-to-run PID controller and SPC monitoring for process disturbance rejection, IIE Trans. **31**, 517–527 (1999)
14.41 N. H. Ruhhal, G. C. Runger, M. Dumitrescu: Control charts and feedback adjustments for a jump disturbance model, J. Qual. Technol. **32**, 379–394 (2000)
14.42 W. H. Woodall: Controversies and contradictions in statistical process control, J. Qual. Technol. **32**, 341–378 (2000)
14.43 H. B. Nembhard: Controlling change: process monitoring and adjustment during transition periods, Qual. Eng. **14**, 229–242 (2001)
14.44 H. B. Nembhard, C. M. Mastrangelo, M.-S. Kao: Statistical monitoring performance for startup operations in a feedback control system, Qual. Reliab. Eng. Int **17**, 379–390 (2001)
14.45 H. B. Nembhard, R. Valverde-Ventura: Cuscore statistics to monitor a non-stationary system. Qual. Reliab. Eng. Int., to appear (2006)
14.46 E. Del Castillo: *Statistical Process Adjustment For Quality Control* (Wiley, New York 2002)

15. Chain Sampling

A brief introduction to the concept of chain sampling is first presented. The chain sampling plan of type ChSP-1 is first reviewed, and a discussion on the design and application of ChSP-1 plans is then presented in the second section of this chapter. Various extensions of chain sampling plans such as the ChSP-4 plan are discussed in the third part. The representation of the ChSP-1 plan as a two-stage cumulative results criterion plan, and its design are discussed in the fourth part. The fifth section relates to the modification of the ChSP-1 plan. The sixth section of this chapter is on the relationship between chain sampling and deferred sentencing plans. A review of sampling inspection plans that are based on the ideas of chain or dependent sampling or deferred sentencing is also made in this section. The economics of chain sampling when compared to quick switching systems is discussed in the seventh section. The eighth section extends the attribute chain sampling to variables inspection. In the ninth section, chain sampling is then compared with the CUSUM approach. The tenth section gives several other interesting extensions of chain sampling, such as chain sampling for mixed attribute and variables inspection. The final section gives concluding remarks.

Acceptance sampling is the methodology that deals with procedures by which decisions to accept or not accept lots of items are based on the results of the inspection of samples. Special purpose acceptance sampling inspection plans (abbreviated to special purpose plans) are tailored for special applications as against general or universal use. Prof. Harold F. Dodge, who is regarded as the father of acceptance sampling, introduced the idea of chain sampling in his 1959 industrial quality control paper [15.1]. Chain sampling can be viewed as a plan based on a cumulative results criterion (CRC), where related batch information is chained or cumulated. The phrase chain sampling is also used in sample surveys to imply snowball sampling for collection of data. It should be noted that this phrase was originally coined in the acceptance sampling literature, and should be distinguished from its usage in other areas.

Chain sampling is extended to two or more stages of cumulation of inspection results with appropriate acceptance criteria for each stage. The theory of chain sampling is also closely related to the various other methods of sampling inspection such as dependent-deferred sentencing, tightened–normal–tightened (TNT) sampling, quick-switching inspection etc.

In this chapter, we provide an introduction to chain sampling and briefly discuss various generalizations of chain sampling plans. We also review a few sampling plans which are related to or based on the methodology of chain sampling. The selection or design of various chain sampling plans is also briefly presented.

15.1 ChSP-1 Chain Sampling Plan

A single-sampling attributes inspection plan calls for acceptance of a lot under consideration if the number of nonconforming units found in a random sample of size n is less than or equal to the acceptance number Ac. Whenever the operating characteristic (OC) curve of a single-sampling plan is required to pass through a prescribed point, the sample size n will be an increasing function of the acceptance number Ac. This fact can be verified from the table of np or unity values given in *Cameron* [15.2] for various values of the probability of acceptance $P_a(p)$ of the lot under consideration whose fraction of nonconforming units is p. The same result is true when the OC curve has to pass through two predetermined points, usually one at the top and the other at the bottom of the OC curve [15.3]. Thus, for situations where small sample sizes are preferred, only single-sampling plans with $Ac = 0$ are desirable [15.4]. However, as observed by *Dodge* [15.1] and several authors, the $Ac = 0$ plan has a pathological OC curve in that the curve starts to drop rapidly even for a very small increase in the fraction nonconforming. In other words, the OC curve of the $Ac = 0$ plan has no point of inflection. Whenever a sampling plan for costly or destructive testing is required, it is common to force the OC curve to pass through a point, say, (LQL, β) where LQL is the limiting quality level for ensuring consumer protection and β is the associated consumer's risk. All other sampling plans, such as double and multiple sampling plans, will require a larger sample size for a one-point protection such as (LQL, β). Unfortunately the $Ac = 0$ plan has the following two disadvantages:

1. The OC curve of the $Ac = 0$ plan has no point of inflection and hence it starts to drop rapidly even for the smallest increase in the fraction nonconforming p.
2. The producer dislikes an $Ac = 0$ plan since a single occasional nonconformity will call for the rejection of the lot.

The chain sampling plan ChPS-1 by *Dodge* [15.1] is an answer to the question of whether anything can be done to improve the pathological shape of the OC curve of a zero-acceptance-number plan. A production process, when in a state of statistical control, maintains a constant but unknown fraction nonconforming p. If a series of lots formed from such a stable process is submitted for inspection, which is known as a type B situation, then the samples drawn from the submitted lots are simply random samples drawn directly from the production process. So, it is logical to allow a single occasional nonconforming unit in the current sample whenever the evidence of good past quality, as demonstrated by the i preceding samples containing no nonconforming units, is available. Alternatively we can chain or cumulate the results of past lot inspections to take a decision on the current lot without increasing the sample size.

The operating procedure of the chain sampling plan of type ChSP-1 is formally stated below:

1. From each of the lots submitted, draw a random sample of size n and observe the number of nonconforming units d.
2. Accept the lot if d is zero. Reject the lot if $d > 1$. If $d = 1$, the lot is accepted provided all the samples of size n each drawn from the preceding i lots are free from nonconforming units; otherwise reject the lot.

Thus the chain sampling plan has two parameters: n, the sample size, and i, the number of preceding sample results chained for making a decision on the current lot. It is also required that the consumer has confidence in the producer, and the producer will deliberately not pass a poor-quality lot taking advantage of the small samples used and the utilization of preceding samples to take a decision on the current lot.

The ChSP-1 plan always accepts the lot if $d = 0$ and conditionally accepts it if $d = 1$. The probability that the preceding i samples of size n are free from nonconforming units is $P_{0,n}^i$. Hence, the OC function is $P_a(p) = P_{0,n} + P_{1,n} P_{0,n}^i$ where $P_{d,n}$ is the probability of getting d nonconforming units in a sample of size n. Figure 15.1 shows the improvement in the shape of the OC curve of the zero-acceptance-number single-sampling plan by the use of chain sampling. *Clark* [15.5] provided a discussion on the OC curves of chain sampling plans, a modification and some applications. *Liebesman* et al. [15.6] argue in favor of chain sampling as the attribute sampling standards have the deficiency for small or fractional acceptance number sampling plans. The authors also provided the necessary tables and examples for the chain sampling procedures. Most text books on statistical quality control also contain a section on chain sampling, and provide some applications.

Soundararajan [15.7] constructed tables for the selection of chain sampling plans for given acceptable quality level (AQL, denoted as p_1), producer's risk α, LQL (denoted as p_2) and β. The plans found from this source are approximate, and a more accurate procedure that also minimizes the sum of actual producer's and

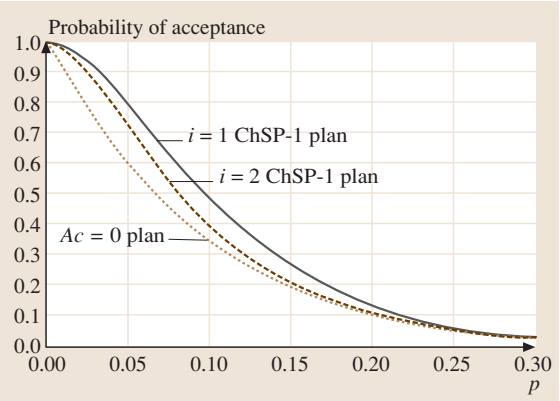


Fig. 15.1 Comparison of OC curves of $Ac = 0$ and ChSP-1 plans

consumer's risks is given by *Govindaraju* [15.8]. Table 15.1, adopted form *Govindaraju* [15.8] is based on the binomial distribution for OC curve of the ChSP-1 plan. This table can also be used to select ChSP-1 plans for given LQL and β only, which may be used in place of zero-acceptance-number plans.

Ohta [15.9] investigated the performance of ChSP-1 plans using the graphical evaluation and review technique (GERT) and derived measures such as OC and average sample number (ASN) for the ChSP-1 plan. *Raju* and *Jothikumar* [15.10] provided a ChSP-1 plan design procedure based on Kullback–Leibler information, and the necessary tables for the selection of the plan. *Govindaraju* [15.11] discussed the design ChSP-1 plan for minimum average total inspection (ATI). There are a number of other sources where the ChSP-1 plan design is discussed. This paper provides additional references on designing chain sampling plans, inter alia, while discussing various extensions and generalizations.

Table 15.1 ChSP-1 plans indexed by AQL and LQL ($\alpha = 0.05$, $\beta = 0.10$) for fraction nonconforming inspection [15.8]. Key $n : i$

LQL	AQL (%)					
(%)	0.1	0.15	0.25	0.40	0.65	1.00
1.5	154:2					
2.0	114:4	124:1				
2.5	91:4	92:2				
3.0	76:3	76:3	82:1			
3.5	65:3	65:3	70:1			
4.0	57:2	57:2	57:2			
4.5	51:2	51:2	51:2			
5.0	45:3	45:3	45:3	49:1		
5.5	41:3	41:3	41:3	45:1		
6.0	38:3	38:2	38:2	38:2		
6.5	35:3	35:2	35:2	35:2		
7.0	32:3	32:3	32:3	32:3		
7.5	30:3	30:3	30:2	30:2		
8.0	28:3	28:3	28:2	28:2	30:1	
8.5	26:3	26:3	26:3	26:3	29:1	
9.0	25:3	25:3	25:2	25:2	27:1	
9.5	24:3	24:3	24:2	24:2	24:2	
10	22:3	22:3	22:3	22:3	22:3	
11	20:3	20:3	20:2	20:2	20:2	
12	19:3	19:3	19:2	19:2	19:2	20:1
13	17:3	17:3	17:3	17:2	17:2	18:1
14	16:3	16:3	16:3	16:2	16:2	16:2
15	15:3	15:3	15:3	15:2	15:2	15:2

15.2 Extended Chain Sampling Plans

Frishman [15.12] extended the ChSP-1 plan and developed ChSP-4 and ChSP-4A plans which incorporate a rejection number greater than 1. Both ChSP-4 and ChSP-4A plans are operated like a traditional double-sampling attributes plan but uses $(k-1)$ past lot results instead of actually taking a second sample from the current lot. The following is a compact tabular representation of Frishman's ChSP-4A plan.

Stage	Sample size	Acceptance number	Rejection number
1	n	a	r
2	$(k\text{-}1)n$	a'	$a'+1$

The ChSP-4 plan restricts r to $a'+1$. The conditional double-sampling plans of *Baker* and *Brobst* [15.13], and the partial and full link-sampling plans of *Harishchandra* and *Srivenkataramana* [15.14] are actually particular cases of the ChSP-4A plan when $k = 2$ and $k = 3$ respectively. However the fact that the OC curves of these plans are the same as the ChSP-4A plan is not reported in both papers [15.15].

Extensive tables for the selection of ChSP-4 and ChSP-4A plans were constructed by *Raju* [15.16, 17] and *Raju* and *Murthy* [15.18–21]. *Raju* and *Jothikumar* [15.22] provided a complete summary of various selection procedures for ChSP-4 and ChSP-4A plans,

and also discussed two further types of optimal plans – the first involving minimum risks and the second based on Kullback–Leibler information. Unfortunately, the tables of Raju et al. for the ChSP-4 or ChSP-4A design require the user to specify the acceptance and rejection numbers. This serious design limitation is not an issue with the procedures and computer programs developed by *Vaerst* [15.23] who discussed the design of ChSP-4A plans involving minimum sample sizes for given AQL, α, LQL and β without assuming any specific acceptance numbers. Raju et al. considered a variety of design criteria while *Vaerst* [15.23] discussed only the (AQL, LQL) criterion. The ChSP-4 and ChSP-4A plans obtained from Raju's tables can be used in any type B situation of a series of lots from a stable production process, not necessarily when the product involves costly or destructive testing. This is because the acceptance numbers covered are above zero. The major disadvantage of *Frishman*'s [15.12] extended ChSP-4 and ChSP-4A plans is that the neighboring lot information is not always utilized. Even though ChSP-4 and ChSP-4A plans require smaller sample sizes than the traditional double-sampling plans, these plans may not be economical compared to other conditional sampling plans.

Bagchi [15.24] presented an extension of the ChSP-1 plan, which calls for additional sampling only when one nonconforming unit is found. The operating procedure of Bagchi's plan is given below:

1. At the outset, inspect n_1 units selected randomly from each lot. Accept the lot if all the n_1 units are conforming; otherwise, reject the lot.
2. If i successive lots are accepted, then inspect only $n_2\,(<n_1)$ items from each of the submitted lots. Accept the lot as long as no nonconforming units are found. If two or more nonconforming units are found, reject the lot. In the event of one nonconforming unit being found in n_2 inspected units, then inspect a further sample (n_1-n_2) units from the same lot. Accept the lot under consideration if no further nonconforming units are found in the additional (n_1-n_2) inspected units; otherwise reject the lot.

Representing Bagchi's plan as a Markov chain, *Subramani* and *Govindaraju* [15.25] derived the steady-state OC function and a few other performance measures.

15.3 Two-Stage Chain Sampling

Dodge and *Stephens* [15.26] viewed the chain sampling approach as a cumulative results criterion (CRC) applied in two stages and extended it to include larger acceptance numbers. Their approach calls for the first stage of cumulation of a maximum of k_1 consecutive lot results, during which acceptance is allowed if the maximum allowable nonconforming units is c_1 or less. After passing the first stage of cumulation (i.e. when k_1 consecutive lots are accepted), the second stage of cumulation of $k_2(>k_1)$ lot results begins. In the second stage of cumulation, an acceptance number of $c_2(>c_1)$ is applied. *Stephens* and *Dodge* [15.27] presented a further generalization of the family of two-stage chain sampling inspection plans by using different sample sizes in the two stages. We state below the complete operating procedure of a generalized two-stage chain sampling plan.

1. At the outset, draw a random sample of n_1 units from the first lot. In general, a sample of size $n_j (j=1,2)$ will be taken while operating in the j^{th} stage of cumulation.
2. Record d, the number of nonconforming units in each sample, as well as D, the cumulative number of nonconforming units from the first up to and including the current sample. As long as $D_i \le c_1 (1 \le i \le k_1)$, accept the i^{th} lot.
3. If k_1 consecutive lots are accepted, continue to cumulate the number of nonconforming units D in the k_1 samples plus additional samples up to but no more than k_2 samples. During this second stage of cumulation, accept the lots as long as $D_i \le c_2 (k_1 < i \le k_2)$.
4. After passing the second stage of k_2 lot acceptances, start cumulation as a moving total over k_2 consecutive samples (by adding the current lot result and dropping the k_2^{th} preceding lot result). Continue to accept lots as long as $D_i \le c_2 (i > k_2)$.
5. If, in any stage of sampling, $D_i > c_i$ then reject the lot and return to Step 1 (a fresh restart of the cumulation procedure).

Figure 15.2 shows how the cumulative results criterion is used in a two-stage chain sampling plan when $k_1 = 3$ and $k_2 = 5$.

An important subset of the generalized two-stage chain sampling plan is when $n_1 = n_2$ and this subset is designated as ChSP-(c_1, c_2); there are five parameters:

n, k_1, k_2, c_1, and c_2. The original chain sampling plan ChSP-1 of *Dodge* [15.1] is a further subset of the ChSP-(0, 1) plan with $k_1 = k_2 - 1$. That is, the OC curve of the generalized two-stage chain sampling plan is equivalent to the OC curve of the ChSP-1 plan when $k_1 = k_2 - 1$. *Dodge* and *Stephens* [15.26] derived the OC function of ChSP-(0, 1) plan as

$$P_a(p) = \frac{P_{0,n}\left(1 - P_{0,n}\right) + P_{0,n}^{k_1} P_{1,n}\left(1 - P_{0,n}^{k_2 - k_1}\right)}{1 - P_{0,n} + P_{0,n}^{k_1} P_{1,n}\left(1 - P_{0,n}^{k_2 - k_1}\right)}\,,$$

$$k_2 > k_1\,.$$

As achieved by the ChSP-1 plan, the ChSP-(0,1) plan also overcomes the disadvantages of the zero-acceptance-number plan. Its operating procedure can be succinctly stated as follows:

1. A random sample of size n is taken from each successive lot, and the number of nonconforming units in each sample is recorded, as well as the cumulative number of nonconforming units found so far.
2. Accept the lot associated with each new sample as long as no nonconforming units are found.
3. Once k_1 lots have been accepted, accept subsequent lots as long as the cumulative number of nonconforming units is no greater than one.
4. Once $k_2 > k_1$ lots have been accepted, cumulate the number of nonconforming units over at most k_2 lots, and continue to accept as long as this cumulative number of nonconforming units is one or none.
5. If, at any stage, the cumulative number of nonconforming units becomes greater than one, reject the current lot and return to Step 1.

Procedures and tables for the design of ChSP-(0,1) plan are available in *Soundararajan* and *Govindaraju* [15.28], and *Subramani* and *Govindaraju* [15.29]. *Govindaraju* and *Subramani* [15.30] showed that the choice of $k_1 = k_2 - 1$ is always forced on the parameters of the ChSP-(0,1) plan when a plan is selected for given AQL, α, LQL, and β. That is, a ChSP-1 plan will be sufficient, and one need not opt for a two-stage cumulation of nonconforming units.

In various technical reports from the Statistics Center at Rutgers University (see *Stephens* [15.31] for a list), *Stephens* and *Dodge* formulated the two-stage chain sampling plan as a Markov chain and evaluated its performance. The performance measures considered by them include the steady-state OC function, ASN and average run length (ARL) etc. For comparison of chain sampling plans with the traditional or noncumulative plans, two types of ARLs are used. The first type of ARL, say ARL_1, is the average number of samples to the first rejection after a sudden shift in the process level, say from p_0 to $p_s (> p_0)$. The usual ARL, say ARL_2, is the average number of samples to the first rejection given the stable process level p_0. The difference $(ARL_1 - ARL_2)$ measures the extra lag due to chain sampling. However, this extra lag may be compensated by gains in sampling efficiency, as explained by *Stephens* and *Dodge* [15.32].

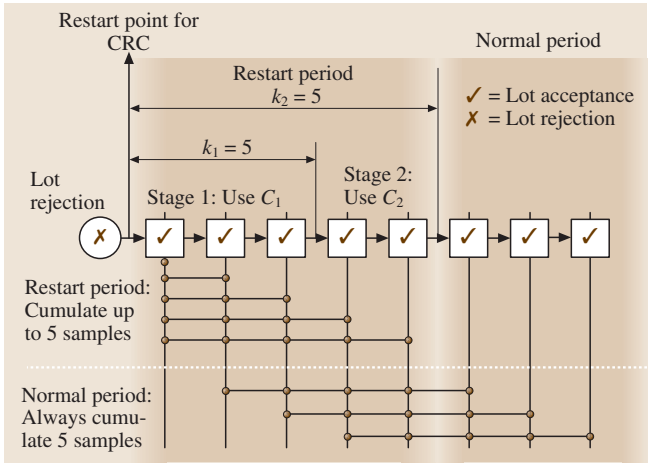


Fig. 15.2 Operation of a two-stage chain sampling plan with $k_1 = 3$ and $k_2 = 5$

Stephens and *Dodge* [15.33] summarized the mathematical approach they have taken to evaluate the performance of some selected two-stage chain sampling plans, while more detailed derivations were published in their technical reports. Based on the expressions for the OC function derived by Stephens and Dodge in their various technical reports (consult *Stephens* [15.31]), *Raju* and *Murthy* [15.34], and *Raju* and *Jothikumar* [15.35] discussed various design procedures for the ChSP-(0,2) and ChSP-(1,2) plans. *Raju* [15.36] extended the two-stage chain sampling to three stages, and evaluated the OC performances of a few selected chain sampling plans, fixing the acceptance numbers for the three stages. The three-stage cumulation procedure becomes very complex, and will play only a limited role for costly or destructive inspections. The three-stage plan will however be useful for general type B lot-by-lot inspections.

15.4 Modified ChSP-1 Plan

In *Dodge*'s [15.1] approach, chaining of past lot results does not always occur. It occurs only when a nonconforming unit is observed in the current sample. This means that the available historical evidence of quality is not fully utilized. *Govindaraju* and *Lai* [15.37] developed a modified chain sampling plan (MChSP-1) that always utilizes the recently available lot-quality history. The operating procedure of the MChSP-1 plan is given below.

1. From each of the submitted lots, draw a random sample of size n. Reject the lot if one or more nonconforming units are found in the sample.
2. Accept the lot if no nonconforming units are found in the sample, provided that the preceding i samples also contained no nonconforming units except in one sample, which may contain at most one nonconforming unit. Otherwise, reject the lot.

A flow chart showing the operation of the MChSP-1 plan is in Fig. 15.3.

The MChSP-1 plan allows a single nonconforming unit in any one of the preceding i samples but the lot under consideration is rejected if the current sample has a nonconforming unit. Thus, the plan gives a psychological protection to the consumer in that it allows acceptance only when all the current sample units are conforming. Allowing one nonconforming unit in any one of the preceding i samples is essential to offer protection to the producer, i.e. to achieve an OC curve with a point of inflection. In the MChSP-1 plan, rejection

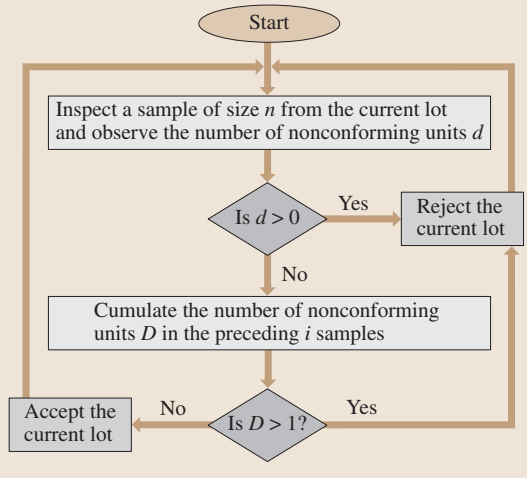


Fig. 15.3 Operation of the MChSP-1 plan

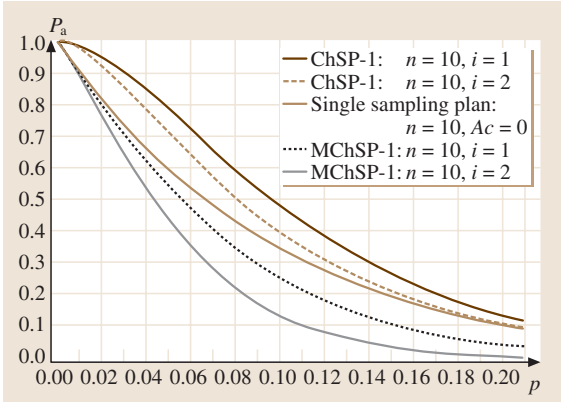


Fig. 15.4 Comparison of OC curves of ChSP-1 and MChSP-1 plans

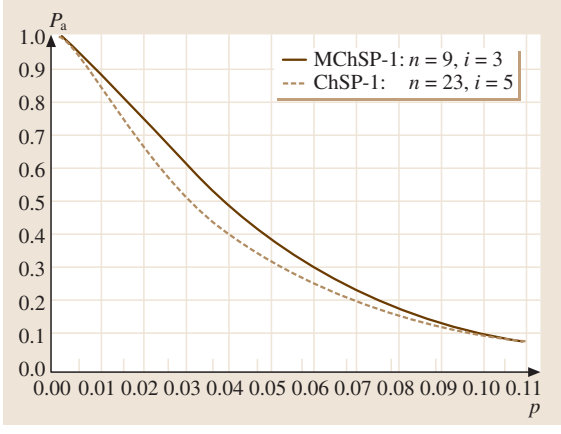


Fig. 15.5 OC curves of matched ChSP-1 and MChSP-1 plans

of lots would occur until the sequence of submissions advances to a stage where two or more nonconforming units were no longer included in the sequence of i samples. In other words, if two or more nonconforming units are found in a single sample, it will result in i subsequent lot rejections. In acceptance sampling, one has to look at the OC curve to have an idea of the protection to the producer as well as to the consumer and what happens in an individual sample or for a few lots is not very important. If two or more nonconforming units are found in a single sample, it does not mean that the subsequent lots need not be inspected since they will be automatically rejected under the proposed plan. It should be noted that results of subsequent lots will be utilized

continuously and the producer has to show an improvement in quality with one or none nonconforming units in the subsequent samples to permit future acceptances. This will act as a strong motivating factor for quality improvement.

The OC function $P_a(p)$ of the MChSP-1 plan was derived by *Govindaraju* and *Lai* [15.37] as $P_a(p) = P_{0,n}(P_{0,n}^i + iP_{0,n}^{i-1}P_{1,n})$. Figure 15.4 compares the OC curves of the ChSP-1 and MChSP-1 plans. From Fig. 15.4, we observe that the MChSP-1 plan decreases the probability of acceptance at poor quality levels but maintains the probability of acceptance at good quality levels when compared to the OC curve of the zero-acceptance-number single-sampling plan. The ChSP-1 plan, on the other hand, increases the probability of acceptance at good quality levels but maintains the probability of acceptance at poor quality levels. To compare the two sampling plans, we need to match them. That is, we need to design sampling plans whose OC curves pass approximately through two prescribed points such as (AQL, 1-α) and (LQL, β). Figure 15.5 gives such a comparison, and establishes that the MChSP-1 plan is efficient in requiring a very small sample size compared to the ChSP-1 plan. A two-stage chain sampling plan would generally require a sample size equal to or more than the sample size of a zero-acceptance single-sampling plan. The MChSP-1 plan will however require a sample size smaller than the zero-acceptance-number plan.

15.5 Chain Sampling and Deferred Sentencing

Like chain sampling plans, there are other plans that use the results of neighboring lots to take a conditional decision of acceptance or rejection. Plans that make use of past lot results are either called chain or dependent sampling plans. Similarly plans that make use of future lot results are known as deferred sentencing plans. These plans have a strategy of accepting the lots conditionally based on the neighboring lot-quality history and are hence referred to as conditional sampling plans. We will briefly review several such conditional sampling plans available in the literature.

In contrast to chain sampling plans, which make use of past lot results, deferred sentencing plans use future lot results. The idea of deferred sentencing was first published in a paper by *Anscombe* et al. [15.38]. The first and simplest type of deferred sentencing scheme [15.38] requires the produced units to be split into small size lots, and one item is selected from each lot for inspection. The lot-sentencing rule is that whenever Y nonconforming units are found out of X or fewer consecutive lots tested, all such clusters of consecutive lots starting from the lot that resulted in the first nonconforming unit to the lot that resulted in the Y^{th} nonconforming unit are rejected. Lots not rejected by this rule are accepted. This rule is further explained in the following sentences. A run of good lots of length X will be accepted at once. If a nonconforming unit occurs, then the lot sentencing or disposition will be deferred until either a further $(X-1)$ lots have been tested or $(Y-1)$ further nonconforming items are found, whichever occurs sooner. At the outset, if the $(X-1)$ succeeding lots result in fewer than $(Y-1)$ nonconforming units, the lot that resulted in the first nonconforming unit and any succeeding lots clear of nonconforming units will be accepted. As soon as Y nonconforming units occur in no more than X lots, all lots not so far sentenced will be rejected. Thus the lot disposition will sometimes be made at once, and sometimes with a delay not exceeding $(X-1)$ lots. Some of the lots to be rejected according to the sentencing rule may already have been rejected through the operation of the rule on a previous cluster of Y nonconforming units that partially overlaps with the cluster being considered. The actual number of new lots rejected under the deferred sentencing rule can be any number from 1 to X. *Anscombe* et al. [15.38] also considered modifications of the above deferred sentencing rule, including inspection of a sample of size more than one from each lot. *Anscombe* et al. [15.38] originally presented their scheme as an alternative to *Dodge*'s [15.39] continuous sampling plan of type CSP-1, which is primarily intended for the partial screening inspection of produced units directly (when lot formation is difficult).

The deferred sentencing idea was formally tailored into an acceptance sampling plan by *Hill* et al. [15.40]. The operating procedure of *Hill* et al. [15.40] scheme is described below:

1. From each lot, select a sample of size n. These lots are accepted as long as no nonconforming units are found in the samples. If one or more nonconforming unit is found, the disposition of the current lot will be deferred until $(X-1)$ succeeding lots are inspected.
2. If the cumulative number of nonconforming units for X consecutive lots is Y or more, then a second sample of size n is taken from each of the lots (beginning with the first lot and ending with the last batch that

showed a nonconforming unit in the sequence of X nonconforming units). If there are less than Y nonconforming units in the X, accept all lots from the first up to, but not including, the next batch that showed a nonconforming unit. The decision on this batch will be deferred until $(X-1)$ succeeding lots are inspected.

Hill et al. [15.40] also evaluated the OC function of some selected schemes and found them to be very economical compared to the traditional sampling plans, including the sequential attribute sampling plan.

Wortham and *Mogg* [15.41] developed a dependent stage sampling (DSSP) plan (DSSP(r, b)), which is operated under steady state as follows:

1. For each lot, draw a sample of size n and observe the number of nonconforming units d.
2. If $d \le r$, accept the lot; if $d > r+b$, reject the lot. If $r+1 \le d \le r+b$, accept the lot if the $(r+b+1-d)^{\text{th}}$ previous lot was accepted; otherwise reject the current lot.

Govindaraju [15.42] observed that the OC function of DSSP(r, b) is the same as the OC function of the repetitive group sampling (RGS) plan of *Sherman* [15.43]. This means that the existing design procedures for the RGS plan can also be used for the design of DSSP(r, b) plan. The deferred state sampling plan of *Wortham* and *Baker* [15.44] has a similar operating procedure except in step 2 in which, when $r+1 \le d \le r+b$, the current lot is accepted if the forthcoming $(r+b+1-d)^{\text{th}}$ lot is accepted. The steady-state OC function of the dependent (deferred) stage sampling plan DSSP(r, b) is given by

$$P_{\text{a}}(p) = \frac{P_{\text{a},r}(p)}{1 - P_{\text{a},r+b}(p) + P_{\text{a},r}(p)}$$

where $P_{\text{a},r}(p)$ is the OC function of the single-sampling plan with acceptance number r and sample size n. Similarly $P_{\text{a},r+b}(p)$ is the OC function of the single-sampling plan with acceptance number $r+b$ and sample size n. A procedure for the determination of the DSSP(r, b) plan for given AQL, α, LQL, and β was also developed by *Vaerst* [15.23].

Wortham and *Baker* [15.45] extended the dependent (deferred) state sampling into a multiple dependent (deferred) state (MDS) plan MDS(r, b, m). The operating procedure of the MDS(r, b, m) plan is given below:

1. For each lot, draw a sample of size n and observe the number of nonconforming units d.
2. If $d \le r$, accept the lot; if $d > r+b$, reject the lot. If $r+1 \le d \le r+b$, accept the lot if the consecutive m preceding lots were all accepted (the consecutive m succeeding lots must be accepted for the deferred MDS(r, b, m) plan).

The steady-state OC function of the MDS(r, b, m) plan is given by the recursive equation

$$P_{\text{a}}(p) = P_{\text{a},r}(p) + \left[P_{\text{a},r+b}(p) + P_{\text{a},r}(p)\right]\left[P_{\text{a}}(p)\right]^m$$

Vaerst [15.46], *Soundararajan* and *Vijayaraghavan* [15.47], *Kuralmani* and *Govindaraju* [15.48], and *Govindaraju* and *Subramani* [15.49] provided detailed tables and procedures for the design of MDS(r, b, m) plans for various requirements.

Vaerst [15.23, 46] modified the MDS(r, b, m) plan to make it on a par with the ChSP-1 plan. The operating procedure of the modified MDS(r, b, m) plan, called MDS-1(r, b, m), is given below:

1. For each lot, draw a sample of size n and observe the number of nonconforming units d.
2. If $d \le r$, accept the lot; if $d > r+b$, reject the lot. If $r+1 \le d \le r+b$, accept the lot if r or fewer nonconforming units are found in each of the consecutive m preceding (succeeding) lots.

When $r=0$, $b=1$, and $m=i$, MDS-1(r, b, m) becomes the ChSP-1 plan. The OC function of the MDS-1(r, b, m) plan is given by the recursive equation

$$P_{\text{a}}(p) = P_{\text{a},r}(p) + \left[P_{\text{a},r+b}(p) + P_{\text{a},r}(p)\right]\left[P_{\text{a},r}(p)\right]^m$$

Vaerst [15.46], *Soundararajan* and *Vijayaraghavan* [15.50], and *Govindaraju* and *Subramani* [15.51] provided detailed tables and procedures for the design of MDS-1(r, b, m) plans for various requirements.

The major and obvious shortcoming of the chain sampling plans is that, since they use sample information from past lots to dispose of the current lot, there is a tendency to reject the current lot of given good quality when the process quality is improving, or to accept the current lot of given bad quality when the process quality is deteriorating. Similar criticisms (in reverse) can be leveled against the deferred sentencing plans. As mentioned earlier, *Stephens* and *Dodge* [15.32] recognizedg this disadvantage of chain sampling and defined the ARL performance measures ARL_1 and ARL_2. Recall that ARL_2 is the average number of lots that will be accepted as a function of the true fraction nonconforming. ARL_1 is the average number of lots accepted after an upward shift in the true fraction nonconforming from the existing level. *Stephens* and *Dodge* [15.52]

evaluated the performance of the two-stage chain sampling plans, comparing the ARLs with matching single- and double-sampling plans having approximately the same OC curve. It was noted that the slightly poorer ARL property due to chaining of lot results is well compensated by the gain in sampling economy. For deferred sentencing schemes, *Hill* et al. [15.40] investigated trends as well as sudden changes in quality. It was found that the deferred sentencing schemes will discriminate better between fairly constant quality at one level and fairly constant quality at another level than will a lot-by-lot plan scheme with the same sample size. However when quality varies considerably from lot to lot, the deferred sentencing scheme was found to operate less satisfactorily, and in certain circumstances the discrimination between good and bad batches may even be worse than for traditional unconditional plans with the same sample size. Furthermore, the deferred sentencing scheme may pose problems of flow, supp1y storage space, and uneven work loads (which is not a problem with chain sampling).

Cox [15.53] provided a more theoretical treatment and considered one-step forward and two-step backward schemes. He represented the lot-sentencing rules as a stochastic process, and applied Bayes's theorem for the sentencing rule. He did recognize the complexity of modeling a multistage procedure. When the submitted lot fraction nonconforming varies, say when a trend exists, both chain and deferred sentencing rules have disadvantages. But this disadvantage can be overcome by combining chain and deferred sentencing rules into a single scheme. This idea was first suggested by *Baker* [15.54] in his dependent deferred state (DDS) plan. *Osanaiye* [15.55] provided a complete methodology of combining chain and deferred sentencing rules, and developed the chain-deferred (ChDP) plan. The ChDP plan has two stages for lot disposition and its operating procedure is given below:

1. From lot number k, inspect n units and count the number of nonconforming units d_k. If $d_k \le c_1$, accept lot number k. If $d_k > c_2$, reject lot numbered k. If $c_1 < d_k \le c_2$, then combine the number of nonconforming units from the immediately succeeding and preceding samples, namely d_{k-1} and d_{k+1}. (Stage 1)
2. If $d_k \le c$, accept the k^{th} lot provided $d_k + d_{k-1} \le c_3$ (chain approach). If $d_k > c$, accept the k^{th} lot provided that $d_k + d_{k+1} \le c_3$ (deferred sentencing).

One possible choice of c is the average of c_1 and $c_3 + 1$. *Osanaiye* [15.55] also provided a comparison of ChDP with the traditional unconditional double-sampling plans as the OC curves of the two types of plans are the same (but the ChDP plan utilizes the neighboring lot results). *Shankar* and *Srivastava* [15.56] and *Shankar* and *Joseph* [15.57] provided a GERT analysis of ChDP plans, following the approach of *Ohta* [15.9]. *Shankar* and *Srivastava* [15.58] discussed the selection of ChDP plans using tables. *Osanaiye* [15.59] provided a multiple-sampling-plan extension of the ChDP plan (called the MChDP plan). MChDP plan uses several neighboring lot results to achieve sampling economy.

Osanaiye [15.60] provided a useful practical discussion on the choice of conditional sampling plans considering autoregressive processes, inert processes (constant process quality shift) and linear trends in quality. Based on a simulation study, it was recommended that the chain-deferred schemes are the cheapest if either the cost of 100% inspection or sampling inspection is high. He recommended the use of the traditional single or double sampling plans only if the opportunity cost of rejected items is very high. *Osanaiye* and *Alebiosu* [15.61] considered the effect of inspection errors on dependent and deferred double-sampling plans vis-a-vis ChDP plans. They observed that the chain-deferred plan in general has a greater tendency to reject nonconforming items than any other plans, irrespective of the magnitude of the inspection error.

Many of the conditional sampling plans, which follow either the approach of chaining or deferring or both, have the same OC curve as a double-sampling (or multiple-sampling) plan. Exploiting this equivalence, *Kuralmani* and *Govindaraju* [15.62] provided a general selection procedure for conditional sampling plans for given AQL and LQL. The plans considered include the conditional double-sampling plan of the ChSP-4A plans of *Frishman* [15.12], the conditional double-sampling plan of *Baker* and *Brobst* [15.13], the link-sampling plan of *Harishchandra* and *Srivenkataramana* [15.14], and the ChDP plan of *Osanaiye* [15.55]. A perusal of the operating ratio LQL/AQL of the tables by *Kuralmani* and *Govindaraju* [15.62] reveals that these conditional sampling plans apply in all type B situations, as a wide range of discrimination between good and bad qualities is provided. However the sample sizes, even though smaller than the traditional unconditional plans, will not be as small as the zero-acceptance-number single-sampling plans. This limits the application of the conditional sampling plans to this special-purpose situation, where the ChSP1 or MChSP-1 plans are most suitable.

Govindaraju [15.63] developed a conditional single-sampling (CSS) plan, which has desirable properties for general applications as well as for costly or destructive

testing. The operating procedure of the CSS plan is as follows.

1. From lot numbered k, select a sample of size n and observe the number of nonconforming units d_k.
2. Cumulate the number of nonconforming units observed for the current lot and the related lots. The related lots will be either past lots, future lots or a combination, depending on whether one is using dependent sampling or deferred sentencing. The lot under consideration is accepted if the total number of nonconforming units in the current lot and the m related lots is less than or equal to the acceptance number, Ac. If d_k is the number of nonconforming units recorded for the k^{th} lot, the rule for the disposition of the k^{th} lot can be stated as:
   a) For dependent or chain single sampling, accept the lot if $d_{k-m} + \cdots + d_{k-1} + d_k \le Ac$; otherwise, reject the lot.
   b) For deferred single sampling, accept the lot if $d_k + d_{k-1} + \cdots + d_{k+m} \le Ac$; otherwise, reject the lot
   c) For dependent-deferred single sampling, where m is desired to be even, accept the lot if $d_{k-\frac{m}{2}} + \cdots + d_k + \cdots + d_{k+\frac{m}{2}} \le Ac$; otherwise, reject the lot.

Thus the CSS plan has three parameters: the sample size n, the acceptance number Ac, and the number of related lot results used, m. As in the case of any dependent sampling procedure, dependent single sampling takes full effect only from the $(m+1)^{\text{st}}$ lot. To maintain equivalent OC protection for the first m lots, an additional sample of mn units can be taken from each lot and the lot be accepted if the total number of nonconforming units is less than or equal to Ac, or additional samples of size $(m+1-i)\,n$ can be taken for the i^{th} lot ($i = 1, 2, \ldots, m$) and the same decision rule be applied. In either case, the results of the additional samples should not be used for lot disposition from lot $(m+1)$. *Govindaraju* [15.63] has shown that the CSS plans require much smaller sample sizes than all other conditional sampling plans. In case of trends in quality, the CSS plan can also be operated as a chain-deferred plan and this will ensure that the changes in lot qualities are somewhat averaged out.

15.6 Comparison of Chain Sampling with Switching Sampling Systems

Dodge [15.64] originally proposed quick-switching sampling (QSS) systems. *Romboski* [15.65] investigated the QSSs and introduced several modifications of the original quick-switching system, which basically consists of two intensities of inspection, say, normal (N) and tightened (T) plans. If a lot is rejected under normal inspection, a switch to tightened inspection will be made; otherwise normal inspection will continue. If a lot is accepted under the tightened inspection, then the normal inspection will be restored; otherwise tightened inspection will be continued. For a review of quick-switching systems, see *Taylor* [15.66] or *Soundararajan* and *Arumainayagam* [15.67].

Taylor [15.66] introduced a new switch number to the original QSS-1 system of *Romboski* [15.65] and compared it with the chain sampling plans. When the sample sizes of normal and tightened plans are equal, the quick-switching systems and the two-stage chain sampling plans were found to give nearly identical performance. Taylor's comparison is only valid for a general situation where acceptance numbers greater than zero are used. For costly or destructive testing, acceptance numbers are kept at zero to achieve minimum sample sizes. In such situations, the chain sampling plans ChSP-1 and ChSP-(0, 1) will fare poorly against other comparable schemes when the incoming quality is at AQL. This fact is explained in the following paragraph using an example.

For costly or destructive testing, a quick-switching system employing zero acceptance number was studied by *Govindaraju* [15.68], and *Soundararajan* and *Arumainayagam* [15.69]. Under this scheme, the normal inspection plan has a sample size of n_{N} units, while the tightened inspection plan has a higher sample size n_{T} $(> n_{\text{N}})$. The acceptance number is kept at zero for both normal and tightened inspection. The switching rule is that a rejection under the normal plan $(n_{\text{N}}, 0)$ will invoke the tightened plan $(n_{\text{T}}, 0)$. An acceptance under the $(n_{\text{T}}, 0)$ plan will revert back to normal inspection. This QSS system, designated as type QSS-$1(n_{\text{N}}, n_{\text{T}}; 0)$, can be used in place of the ChSP-1 and ChSP(0,1) plans. Let AQL $= 1\%$, $\alpha = 5\%$, LQL $= 15\%$, and $\beta = 10\%$. The ChSP-1 plan for the prescribed AQL and LQL conditions is found to be $n = 15$ and $i = 2$ (Table 15.1). The matching QSS-1 system for the prescribed AQL and LQL conditions can be found to be

QSS-1($n_N = 5, n_T = 19$) from the tables given in *Govindaraju* [15.68] or *Kuralmani* and *Govindaraju* [15.70]. At good quality levels, the normal inspection plan will require sampling only five units. Only at poor quality levels, 19 units will be sampled under the QSS system. So, it is obvious that *Dodge*'s [15.1] chain sampling approach is not truly economical at good quality levels but fares well at poor quality levels. However, if the modified chain sampling plan MChSP-1 by *Govindaraju* and *Lai* [15.37] is used, then the sample size needed will only be three units (and i, the number of related lot results to be used, is fixed at seven or eight).

A more general two-plan system having zero acceptance number for the tightened and normal plans was studied by *Calvin* [15.71], *Soundararajan* and *Vijayaraghavan* [15.72], and *Subramani* and *Govindaraju* [15.73]. Calvin's TNT scheme uses zero acceptance numbers for normal and tightened inspection and employs the switching rules of MIL-STD-105 D [15.74], which is also roughly employed in ISO 2859-1:1989 [15.75]. The operating procedure of the TNT scheme, designated TNT (n_N, n_T; $Ac = 0$), is given below:

1. Start with the tightened inspection plan ($n_T, 0$). Switch to normal inspection (Step 2) when t lots in a row are accepted; otherwise continue with the tightened inspection plan.
2. Apply the normal inspection plan ($n_N, 0$). Switch to the tightened plan if a lot rejection is followed by another lot rejection within the next s lots.

Using the tables of *Soundararajan* and *Vijayaraghavan* [15.76], the zero-acceptance-number TNT(n_N, n_T; 0) plan for given AQL = 1%, α = 5%, LQL = 15%, and $\beta = 10\%$ is found to be TNT($n_N = 5, n_T = 16$; $Ac = 0$). We again find that the MChSP-1 plan calls for a smaller sample size when compared to Calvin's zero-acceptance-number TNT plan.

The skip-lot sampling plans of *Dodge* [15.77] and *Perry* [15.78] are based on skipping of sampling inspection of lots on the evidence of good quality history. For a detailed discussion of skip-lot sampling, *Stephens* [15.31] may be consulted. In the skip-lot sampling plan of type SkSP-2 by *Perry* [15.78], once m successive lots are accepted under the reference plan, the chosen reference sampling plan is applied only for a fraction f of the time. *Govindaraju* [15.79] studied the employment of the zero-acceptance-number plan as a reference plan (among several other reference sampling plans) in the skip-lot context. For given AQL = 1%, $\alpha = 5\%$, LQL = 15%, and $\beta = 10\%$, the SkSP-2 plan with a zero-acceptance-number reference plan is found to be $n = 15$ $m = 6$, and $f \simeq 1/5$. Hence the matching ChSP-1 plan $n = 15$ and $i = 2$ is not economical at good quality levels when compared to the SkSP-2 plan $n = 15$, $m = 6$, and $f \simeq 1/5$. This is because the SkSP-2 plan requires the zero-acceptance-number reference plan with a sample size of 15 to be applied only to one in every five lots submitted for inspection once six consecutive lots are accepted under the reference single-sampling plan ($n = 10$, $Ac = 0$). However, the modified MChSP-1 plan is more economical at poor quality levels when compared to the SkSP-2 plan. Both plans require about the same sampling effort at good quality levels.

15.7 Chain Sampling for Variables Inspection

Govindaraju and *Balamurali* [15.80] extended the idea of chain sampling to sampling inspection by variables. This approach is particularly useful when testing is costly or destructive provided the quality variable is measurable on a continuous scale. It is well known that variables plans do call for very low sample sizes when compared to the attribute plans. However not all variables plans possess a satisfactory OC curve, as shown by *Govindaraju* and *Kuralmani* [15.81]. Often, a variables plan is unsatisfactory if the acceptability constant is too large, particularly when the sample size is small. Only in such cases is it necessary to follow the chain sampling approach to improve upon the OC curve of the variables plan. Table 15.2 is useful for deciding whether a given variables sampling plan has a satisfactory OC curve or not. If the acceptability constant k_σ of a known sigma variables plan exceeds $k_{\sigma l}$ then the plan is deemed to have an unsatisfactory OC curve, like an $Ac = 0$ attributes plan.

The operating procedure of the chain sampling plan for variables inspection is as follows:

1. Take a random sample of size n_σ, say $(x_1, x_2, ..., x_{n_\sigma})$ and compute

$$v = \left(\frac{U - \bar{X}}{\sigma}\right), \text{ where } \bar{X} = \frac{1}{n_\sigma}\sum_{i=1}^{n_\sigma} x_i .$$

Table 15.2 Limits for deciding unsatisfactory variables plans

n_σ	$k_{\sigma l}$	n_σ	$k_{\sigma l}$	n_σ	$k_{\sigma l}$	n_σ	$k_{\sigma l}$
1	0	16	2.3642	31	3.3970	46	4.1830
2	0.4458	17	2.4465	32	3.4549	47	4.2302
3	0.7280	18	2.5262	33	3.5119	48	4.2769
4	0.9457	19	2.6034	34	3.5680	49	4.3231
5	1.1278	20	2.6785	35	3.6232	50	4.3688
6	1.2869	21	2.7515	36	3.6776	51	4.4140
7	1.4297	22	2.8227	37	3.7312	52	4.4588
8	1.5603	23	2.8921	38	3.7841	53	4.5032
9	1.6812	24	2.9599	39	3.8362	54	4.5471
10	1.7943	25	3.0262	40	3.8876	55	4.5905
11	1.9009	26	3.0910	41	3.9384	56	4.6336
12	2.0020	27	3.1546	42	3.9885	57	4.6763
13	2.0983	28	3.2169	43	4.0380	58	4.7186
14	2.1904	29	3.2780	44	4.0869	59	4.7605
15	2.2789	30	3.3380	45	4.1352	60	4.8021

2. Accept the lot if $v \geq k_\sigma$ and reject if $v < k'_\sigma$. If $k'_\sigma \leq v < k_\sigma$, accept the lot provided the preceding i lots were accepted on the condition that $v \geq k_\sigma$.

Thus the variables chain sampling plan has four parameters: the sample size n_σ, the acceptability constants k_σ and k'_σ $(< k_\sigma)$, and i, the number of preceding lots used for conditionally accepting the lot. The OC function of this plan is given by $P_a(p) = P_V + (P'_V - P_V)P_V^i$, where $P_V = \Pr(v \geq k_\sigma)$ is the probability of accepting the lot under the variables plan (n_σ, k_σ) and $P'_V = \Pr(v \geq k'_\sigma)$ is the probability of accepting the lot under the variables plan (n_σ, k'_σ). Even though the above operating procedure of the variables chain sampling plan is of general nature, it would be appropriate to fix $k'_\sigma = k_{\sigma l}$. For example, suppose that a variables plan with $n_\sigma = 5$ and $k_\sigma = 2.46$ is currently under use. From Table 15.2, the limit for the undesirable acceptability constant $k_{\sigma l}$ for $n_\sigma = 5$ is obtained as 1.1278. As the actual acceptability constant $k_\sigma (= 2.26)$ is greater than $k_{\sigma l} (= 1.1278)$, the variables plan can be declared to possess an unsatisfactory OC curve. Hence it is desirable to chain the results of neighboring lots to improve upon the shape of the OC curve of the variables plan $n_\sigma = 5$ and $k_\sigma = 2.46$. That is, the variables plan currently under use with $n_\sigma = 5$ and $k_\sigma = 2.46$ will be operated as a chain sampling plan fixing $i = 4$. A more detailed procedure on designing chain sampling for variables inspection, including the case when sigma is unknown, is available in *Govindaraju* and *Balamurali* [15.80]. The chain sampling for variables will be particularly useful when inspection costs are prohibitively high, and the quality characteristic is measurable on a continuous scale.

15.8 Chain Sampling and CUSUM

In this section, we will discuss some of the interesting relationships between the cumulative sum (CUSUM) approach of *Page* [15.82, 83] and the chain sampling approach of *Dodge* [15.1]. The CUSUM approach is largely popular in the area of statistical process control (SPC) but *Page* [15.82] intended it for use in acceptance sampling as well. *Page* [15.82] compares his CUSUM-based inspection scheme with the deferred sentencing schemes of *Anscombe* et al. [15.38], and the continuous sampling plan CSP-1 of *Dodge* [15.39] to evaluate their relative performance. In fact Dodge's CSP-1 plan forms the theoretical basis for his ChSP-1 chain sampling plan. A more formal acceptance sampling scheme based on the one-sided CUSUM for lot-by-lot inspection was proposed by *Beattie* [15.84]. Beattie's plan calls for drawing a random sample of size n from each lot and observing the number of nonconforming units d. For each lot, a CUSUM value is calculated for a given

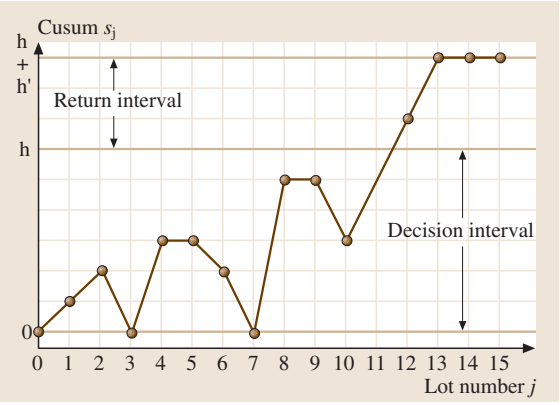


Fig. 15.6 Beattie's CUSUM acceptance sampling plan

slack parameter k. If the computed CUSUM is within the decision interval $(0, h)$, then the lot is accepted. If the CUSUM is within the return interval $(h, h+h')$, then the lot is rejected. If the CUSUM falls below zero, it is reset to zero. Similarly if the CUSUM exceeds $h+h'$, it is reset to $h+h'$. In other words, for the j-th lot, the plotted CUSUM can be succinctly defined as $S_j = \text{Min}\{h+h',\ \text{Max}\{(d_j - k) + S_{j-1}, 0\}\}$ with $S_0 = 0$. Beattie's plan is easily implemented using the typical number of nonconforming units CUSUM chart for lot-by-lot inspection Fig. 15.6. *Prairie* and *Zimmer* [15.85] provided detailed tables and nomographs for the selection of Beattie's CUSUM acceptance sampling plan. An application is also reported in [15.86].

Beattie [15.87] introduced a two-stage semi-continuous plan where the CUSUM approach is followed, and the product is accepted as long as the CUSUM, S_j, is within the decision interval $(0, h)$. For product falling in the return interval $(h, h+h')$, an acceptance sampling plan such as the single- or double-sampling plan is used to dispose of the lots. *Beattie* [15.87] compared the two-stage semi-continuous plan with the ChSP-4A plan of *Frishman* [15.12] and the deferred sentencing scheme of *Hill* et al. [15.40]. Beattie remarked that chain sampling plans (ChSP-4A type) call for a steady rate of sampling and are simple to administer. The two-stage semi-continuous sampling plan achieved some gain in the average sample number at good quality levels, but it is more difficult to administer. The two-stage semi-continuous plan also requires a larger sample size than the ChSP-4A plans when the true quality is poorer than acceptable levels.

We will now explore an interesting equivalence between the ChSP-1 plan, and a CUSUM scheme intended for high-yield or low-fraction-nonconforming production processes for which the traditional p or np control charts are not useful. *Lucas* [15.88] gave a signal rule for lack of statistical control if there are two or more counts within an interval of t samples. In the case of a process with a low fraction nonconforming, this means that, if two or more nonconforming units are observed in any t consecutive samples or less, a signal for an upward shift in the process fraction level is obtained. It should be noted that, if two or more nonconforming units are found even in the same sample, a signal for lack of statistical control will be obtained. *Govindaraju* and *Lai* [15.89] discuss the design of *Lucas*'s [15.88] scheme, and provided a method of obtaining the parameters n (the subgroup or sample size) and t (the maximum number of consecutive samples considered for a signal).

Lucas [15.88] has shown that his signal rule is equivalent to a CUSUM scheme having a reference value k of $1/t$ and decision interval $h = 1$ for detecting an increase in the process count level. It was also shown that a fast initial response (FIR) feature can be added to the CUSUM scheme (see *Lucas* and *Crosier* [15.90]) with an additional sub-rule that signals lack of statistical control if the first count occurs before the t-th sample. This FIR CUSUM scheme has a head start of $S_0 = 1 - k$ with $k = 1/t$ and $h = 1$. Consider the ChSP-1 plan of *Dodge* [15.1], which rejects a lot if two or more counts (of nonconformity or nonconforming units) occur but allows acceptance of the lot if no counts occur or a single count is preceded by t (the symbol i was used before) lots having samples with no counts. If the decision to reject a lot is translated as the decision of declaring the process to be not in statistical control, then it is seen that Lucas's scheme and the ChSP-1 plan are the same. This equivalence will be even clearer if one considers the operation of the two-stage chain sampling plan ChSP(0,1) of *Dodge* and *Stephens* [15.26] given in Sect. 15.3. When $k_2 = k_1 + 1$, the ChSP(0,1) plan is equivalent to the ChSP-1 plan with $t = k_1$. So it can also be noted that the sub-rule of not allowing any count for the first t samples suggested for the FIR CUSUM scheme of *Lucas* [15.88] is an inherent feature of the two-stage chain sampling scheme. This means that the ChSP-1 plan is equivalent to the FIR CUSUM scheme with the head start of $(1-k)$ with $k = 1/t$ and $h = 1$.

15.9 Other Interesting Extensions

Mixed sampling plans are two-phase sampling plans in which both variable quality characteristics and attribute quality measures are used in deciding the acceptance or rejection of the lot. *Baker* and *Thomas* [15.91] reported the application of chain sampling for acceptance testing for armor packages. Their procedure uses chain sampling for testing structural integrity (attributes inspection) while a variables sampling plan is used for testing penetration-depth quality characteristic. The authors also suggested the simultaneous use of control charts along with their proposed acceptance sampling procedures. *Suresh* and *Devaarul* [15.92] proposed a more formal mixed acceptance sampling plan where a chain sampling plan is used for the attribute phase. *Suresh* and *Devaarul* [15.92] also obtained the OC function for their mixed plan, and discussed various selection procedures. To control multidimensional characteristics, *Suresh* and *Devaarul* [15.93] developed multidimensional mixed sampling plans (MDMSP). These plans handles several quality characteristics during the variable phase of the plan, while the attribute sampling phase can be based on chain sampling or other attribute plans.

In some situations it is desirable to adopt three attribute classes, where items are classified into three categories: good, marginal and bad [15.94]. *Shankar* et al. [15.95] developed three-class chain sampling plans and derived various performance measures through the GERT approach and also discussed their design.

Suresh and *Deepa* [15.96] provided a discussion on formulating a chain sampling plan given a prior gamma or beta distribution for product quality. Tables for the selection of the plans and examples are also provided by *Suresh* and *Deepa* [15.96]. This approach will further improve the sampling efficiency of chain sampling plans.

15.10 Concluding Remarks

This chapter largely reviews the methodology of chain sampling for lot-by-lot inspection of quality. Various extensions of the original chain sampling plan ChSP-1 of *Dodge* [15.1] and modifications are briefly reviewed. The chain sampling approach is primarily useful for costly or destructive testing, where small sample sizes are preferred. As chain sampling plans achieve greater sampling economy, these are combined with the approach of deferred sentencing so that the combined plan can be used for any general situation. This chapter does not cover design of chain sampling plans in any great detail. One may consult textbooks such as *Schilling* [15.97] or *Stephens* [15.31,98] for detailed tables. A large number of papers primarily dealing with the design of chain sampling plans are available only in journals, and some of them are listed as references. It is often remarked that designing sampling plans is more of an art than a science. There are statistical, engineering and other administrative aspects to be taken into account for successful implementation of any sampling inspection plan, including chain sampling plans. For example, for administrative and other reasons, the sample size may be fixed. Given this limitation, which sampling plan should be used requires careful consideration. Several candidate sampling plans, including chain sampling plans, must first be sought, and then the selection of a particular type of plan must be made based on performance measures such as the OC curve etc. The effectiveness of the chosen plan or sampling scheme must be monitored over time, and changes made if necessary.

References

15.1 H. F. Dodge: Chain sampling inspection plan, Indust. Qual. Contr. **11**, 10–13 (1955) (originally presented on the program of the Annual Middle Atlantic Regional Conference, American Society for Quality Control, Baltimore, MD, February 5, 1954; also reproduced in J. Qual. Technol. **9** p. 139–142 (1997))

15.2 J. M. Cameron: Tables for constructing, for computing the operating characteristics of single-sampling plans, Ind. Qual. Contr. **9**, 37–39 (1952)

15.3 W. C. Guenther: Use of the binomial, hypergeometric, Poisson tables to obtain sampling plans, J. Qual. Technol. **1**, 105–109 (1969)

15.4 G. J. Hahn: Minimum size sampling plans, J. Qual. Technol. **6**, 121–127 (1974)

15.5 C. R. Clark: O-C curves for ChSP-1 chain sampling plans, Ind. Qual. Contr. **17**, 10–12 (1960)

15.6 B. S. Liebesman, F. C. Hawley, H. M. Wadsworth: Reviews of standards, specifications: small acceptance number plans for use in military standard 105D, J. Qual. Technol. **16**, 219–231 (1984)

15.7 V. Soundararajan: Procedures, tables for construction, selection of chain sampling plans (ChSP-1), J. Qual. Technol. **10**, 56–60 and 99–103 (1978)

15.8 K. Govindaraju: Selection of ChSP-1 chain sampling plans for given acceptable quality level and limiting quality level, Commun. Stat. Theor. Methods **19**, 2179–2190 (1990)

15.9 H. Ohta: GERT analysis of chain sampling inspection plans, Bull. Uni. Osaka Prefecture Sec. A Eng. Nat. Sci. **27**, 167–174 (1979)

15.10 C. Raju, J. Jothikumar: A design of chain sampling plan ChSP-1 based on Kullback–Leibler information, J. Appl. Stat. **21**, 153–160 (1994)

15.11 K. Govindaraju: Selection of minimum ATI ChSP-1 chain sampling plans, IAPQR Trans. **14**, 91–97 (1990)

15.12 F. Frishman: An extended chain sampling inspection plan, Ind. Qual. Contr. **17**, 10–12 (1960)

15.13 R. C. Baker, R. W. Brobst: Conditional double sampling, J. Qual. Technol. **10**, 150–154 (1978)

15.14 K. Harishchandra, T. Srivenkataramana: Link sampling for attributes, Commun. Stat. Theor. Methods **11**, 1855–1868 (1982)

15.15 C. Raju: On equivalence of OC functions of certain conditional sampling plans, Commun. Stat.-Simul. C. **21**, 961–969 (1992)

15.16 C. Raju: Procedures and tables for the construction and selection of chain sampling plans ChSP-4A(c_1,c_2) r, Part 1, J. Appl. Stat. **18**, 361–381 (1991)

15.17 C. Raju: Procedures and tables for the construction and selection of chain sampling plans ChSP-4A(c_1,c_2) r, Part 2, J. Appl. Stat. **19**, 125–140 (1992)

15.18 C. Raju, M. N. N. Murthy: Procedures and tables for the construction and selection of chain sampling plans ChSP-4A(c_1,c_2) r, Part 3, J. Appl. Stat. **20**, 495–511 (1993)

15.19 C. Raju, M. N. N. Murthy: Procedures and tables for the construction and selection of chain sampling plans ChSP-4 (c_1,c_2) – Part 4, J. Appl. Stat. **22**, 261–271 (1995)

15.20 C. Raju, M. N. N. Murthy: Minimum risks chain sampling plans ChSP$-4(c_1,c_2)$ indexed by acceptable quality level and limiting quality level, J. Appl. Stat. **22**, 389–426 (1995)

15.21 C. Raju, M. N. N. Murthy: Designing ChSP-4(c_1,c_2) plans, J. Appl. Stat. **21**, 261–27 (1994)

15.22 C. Raju, J. Jothikumar: Procedures and tables for the construction and selection of chain sampling plans ChSP-4 (c_1,c_2) r–Part 5, J. Appl. Stat. **24**, 49–76 (1997)

15.23 R. Vaerst: *About the Determination of Minimum Conditional Attribute Acceptance Sampling Procedures, Dissertation* (Univ. Siegen, Siegen 1981)

15.24 S. B. Bagchi: An extension of chain sampling plan, IAPQR Trans. **1**, 19–22 (1976)

15.25 K. Subramani, K. Govindaraju: Bagchi's extended two stage chain sampling plan, IAPQR Trans. **19**, 79–83 (1994)

15.26 H. F. Dodge, K. S. Stephens: Some new chain sampling inspection plans, Ind. Qual. Contr. **23**, 61–67 (1966)

15.27 K. S. Stephens, H. F. Dodge: Two-stage chain sampling inspection plans with different sample sizes in the two stages, J. Qual. Technol. **8**, 209–224 (1976)

15.28 V. Soundararajan, K. Govindaraju: Construction and selection of chain sampling plans ChSP-(0, 1), J. Qual. Technol. **15**, 180–185 (1983)

15.29 K. Subramani, K. Govindaraju: Selection of ChSP-(0,1) plans for given IQL, MAPD, Int. J. Qual. Rel. Man. **8**, 39–45 (1991)

15.30 K. Govindaraju, K. Subramani: Selection of chain sampling plans ChSP-1, ChSP-(0,1) for given acceptable quality level and limiting quality level, Am. J. Math. Man. Sci. **13**, 123–136 (1993)

15.31 K. S. Stephens: How to Perform Skip-lot and Chain sampling. In: *ASQ Basic References in Quality Control*, Vol. 4, ed. by E. F. Mykytka (Am. Soc. Quality Control, Wisconsin 1995)

15.32 K. S. Stephens, H. F. Dodge: *Evaluation of response characteristics of chain sampling inspection plans*, Technical Report N-25 (Rutgers, Piscataway 1967)

15.33 K. S. Stephens, H. F. Dodge: An application of Markov chains for the evaluation of the operating characteristics of chain sampling inspection plans, The QR Journal **1**, 131–138 (1974)

15.34 C. Raju, M. N. N. Murthy: Two-stage chain sampling plans ChSP-(0,2), ChSP-(1,2)–Part 1, Commun. Stat. Simul. C **25**, 557–572 (1996)

15.35 J. Jothikumar, C. Raju: Two stage chain sampling plans ChSP-(0,2), ChSP-(1,2)–Part 2, Commun. Stat. Simul. C **25**, 817–834 (1996)

15.36 C. Raju: Three-stage chain sampling plans, Commun. Stat. Theor. Methods **20**, 1777–1801 (1991)

15.37 K. Govindaraju, C. D. Lai: A Modified ChSP-1 chain sampling plan, MChSP-1 with very small sample sizes, Amer. J. Math. Man. Sci. **18**, 343–358 (1998)

15.38 F. J. Anscombe, H. J. Godwin, R. L. Plackett: Methods of deferred sentencing in testing the fraction defective of acontinuous output, J. R. Stat. Soc. Suppl. **9**, 198–217 (1947)

15.39 H. F. Dodge: A sampling inspection plan for continuous production, Ann. Math. Stat. **14**, 264–279 (1943) also in J. Qual. Technol. **9**, p. 120–124 (1977)

15.40 I. D. Hill, G. Horsnell, B. T. Warner: Deferred sentencing schemes, Appl. Stat. **8**, 86–91 (1959)

15.41 A. W. Wortham, J. M. Mogg: Dependent stage sampling inspection, Int. J. Prod. Res. **8**, 385–395 (1970)

15.42 K. Govindaraju: An interesting observation in acceptance sampling, Econ. Qual. Contr. **2**, 89–92 (1987)

15.43 R. E. Sherman: Design, evaluation of repetitive group sampling plan, Technometrics **7**, 11–21 (1965)

15.44 A. W. Wortham, R. C. Baker: Deferred state sampling procedures, Ann. Assur. Sci. , 64–70 (1971) 1971 Symposium on Reliability

15.45 A. W. Wortham, R. C. Baker: Multiple deferred state sampling inspection, Int. J. Prod. Res. **14**, 719–731 (1976)

15.46 R. Vaerst: A method to determine MDS sampling plans, Methods Oper. Res. **37**, 477–485 (1980) (in German)

15.47 V. Soundararajan, R. Vijayaraghavan: Construction, selection of multiple dependent (deferred) state sampling plan, J. Appl. Stat. **17**, 397–409 (1990)

15.48 V. Kuralmani, K. Govindaraju: Selection of multiple deferred (dependent) state sampling plans, Commun. Stat. Theor. Methods **21**, 1339–1366 (1992)

15.49 K. Govindaraju, K. Subramani: Selection of multiple deferred (dependent) state sampling plans for given acceptable quality level and limiting quality level, J. Appl. Stat. **20**, 423–428 (1993)

15.50 V. Soundararajan, R. Vijayaraghavan: On designing multiple deferred state sampling [MDS-1(0, 2)] plans involving minimum risks, J. Appl. Statist. **16**, 87–94 (1989)

15.51 K. Govindaraju, K. Subramani: Selection of multiple deferred state MDS-1 sampling plans for given acceptable quality level and limiting quality level involving minimum risks, J. Appl. Stat. **17**, 427–434 (1990)

15.52 K. S. Stephens, H. F. Dodge: Comparison of chain sampling plans with single and double sampling plans, J. Qual. Technol. **8**, 24–33 (1976)

15.53 D. R. Cox: Serial sampling acceptance schemes derived from Bayes's Theorem, Technometrics **2**, 353–360 (1960)

15.54 R. C. Baker: *Dependent-Deferred State Attribute Acceptance Sampling, Dissertation* (A & M Univ. College Station, Texas 1971)

15.55 P. A. Osanaiye: Chain-deferred inspection plans, Appl. Stat. **32**, 19–24 (1983)

15.56 G. Shankar, R. K. Srivastava: GERT analysis of two-stage deferred sampling plan, Metron **54**, 181–193 (1996)

15.57 G. Shankar, S. Joseph: GERT analysis of chain-deferred (ChDF-2) sampling plan, IAPQR Trans. **21**, 119–124 (1996)

15.58 G. Shankar, R. K. Srivastava: Procedures and tables for construction and selection of chain-deferred (ChDF-2) sampling plan, Int. J. Man. Syst. **12**, 151–156 (1996)

15.59 P. A. Osanaiye: Multiple chain-deferred inspection plans, their compatibility with the multiple plans in MIL-STD-105D and equivalent schemes, J. Appl. Stat. **12**, 71–81 (1985)

15.60 P. A. Osanaiye: An economic choice of sampling inspection plans under varying process quality, Appl. Stat. **38**, 301–308 (1989)

15.61 P. A. Osanaiye: Effects of industrial inspection errors on some plans that utilise the surrounding lot information, J. Appl. Stat. **15**, 295–304 (1988)

15.62 V. Kuralmani, K. Govindaraju: Selection of conditional sampling plans for given AQL and LQL, J. Appl. Stat. **20**, 467–479 (1993)

15.63 K. Govindaraju: Conditional single sampling procedure, Commun. Stat. Theor. Methods **26**, 1215–1226 (1997)

15.64 H. F. Dodge: *A new dual system of acceptance sampling*, Technical Report No. 16 (Rutgers, Piscataway 1967)

15.65 L. D. Romboski: *An Investigation of Quick Switching Acceptance Sampling Systems, Dissertation* (Rutgers, Piscataway 1969)

15.66 W. A. Taylor: Quick switching systems, J. Qual. Technol. **28**, 460–472 (1996)

15.67 V. Soundararajan, S. D. Arumainayagam: Construction, selection of modified quick switching systems, J. Appl. Stat. **17**, 83–114 (1990)

15.68 K. Govindaraju: Procedures and tables for the selection of zero acceptance number quick switching system for compliance sampling, Commun. Stat. Simul. **C20**, 157–172 (1991)

15.69 V. Soundararajan, S. D. Arumainayagam: Quick switching system for costly, destructive testing, Sankhya Series B **54**, 1–12 (1992)

15.70 V. Kuralmani, K. Govindaraju: Modified tables for the selection of quick switching systems for agiven (AQL, LQL), Commun. Stat. Simul. C. **21**, 1103–1123 (1992)

15.71 T. W. Calvin: TNT zero acceptance number sampling, ASQC Tech. Conf. Trans , 35–39 (1977)

15.72 V. Soundararajan, R. Vijayaraghavan: Construction and selection of tightened-normal-tightened (TNT) plans, J. Qual. Technol. **22**, 146–153 (1990)

15.73 K. Subramani, K. Govindaraju: Selection of zero acceptance number tightened-normal-tightened scheme for given (AQL, LQL), Int. J. Man. Syst. **10**, 13–120 (1994)

15.74 MIL-STD-105 D: *Sampling Procedures and Tables for Inspection by Attributes* (US Government Printing Office, Washington, DC 1963)

15.75 ISO 2859-1: *1989 Sampling Procedures for Inspection by Attributes—Part 1: Sampling Plans Indexed by Acceptable Quality Level (AQL) for Lot-by-Lot Inspection* (International Standards Organization, Geneva 1989)

15.76 V. Soundararajan, R. Vijayaraghavan: Construction, selection of tightened-normal-tightened sampling inspection scheme of type TNT-$(n_1, n_2; c)$, J. Appl. Stat. **19**, 339–349 (1992)

15.77 H. F. Dodge: Skip-lot sampling plan, Ind. Qual. Contr. **11**, 3–5 (1955) (also reproduced in J. Qual. Technol. **9** 143-145 (1977))

15.78 R. L. Perry: Skip-lot sampling plans, J. Qual. Technol. **5**, 123–130 (1973)

15.79 K. Govindaraju: *Contributions to the Study of Certain Special Purpose Plans, Dissertation* (Univ. Madras, Madras 1985)

15.80 K. Govindaraju, S. Balamurali: Chain sampling for variables inspection, J. Appl. Stat. **25**, 103–109 (1998)

15.81 K. Govindaraju, V. Kuralmani: A note on the operating characteristic curve of the known sigma single sampling variables plan, Commun. Stat. Theor. Methods **21**, 2339–2347 (1992)

15.82 E. S. Page: Continuous inspection schemes, Biometrika **41**, 100–115 (1954)

15.83 E. S. Page: Cumulative sum charts, Technometrics **3**, 1–9 (1961)

15.84 D. W. Beattie: Acontinuous acceptance sampling procedure based upon acumulative sum chart for the number of defectives, Appl. Stat. **11**, 137–147 (1962)

15.85 R. R. Prairie, W. J. Zimmer: Graphs, tables and discussion to aid in the design and evaluation of an acceptance sampling procedure based on cumulative sums, J. Qual. Technol. **5**, 58–66 (1973)

15.86 O. M. Ecker, R. S. Elder, L. P. Provost: Reviews of standards, specifications: United States Department of Agriculture CUSUM acceptance sampling procedures, J. Qual. Technol. **13**, 59–64 (1981)

15.87 D. W. Beattie: Patrol inspection, Appl. Stat. **17**, 1–16 (1968)

15.88 J. M. Lucas: Control schemes for low count levels, J. Qual. Technol. **21**, 199–201 (1989)

15.89 K. Govindaraju, C. D. Lai: Statistical design of control schemes for low fraction nonconforming, Qual. Eng. **11**, 15–19 (1998)

15.90 J. M. Lucas, R. B. Crosier: Fast initial response (FIR) for cumulative sum quality control schemes, Technometrics **24**, 199–205 (1982)

15.91 W. Baker, J. Thomas: Armor acceptance procedure, Qual. Eng. **5**, 213–223 (1992)

15.92 K. K. Suresh, S. Devaarul: Designing, selection of mixed sampling plan with chain sampling as attribute plan, Qual. Eng. **15**, 155–160 (2002)

15.93 K. K. Suresh, S. Devaarul: Multidimensional mixed sampling plans, Qual. Eng. **16**, 233–237 (2003)

15.94 D. F. Bray, D. A. Lyon, J. W. Burr: Three class attribute plans in acceptance sampling, Technometrics **15**, 575–58 (1973)

15.95 S. Joseph, G. Shankar, B. N. Mohapatra: Chain sampling plan for three attribute classes, Int. J. Qual. Reliab. Man. **8**, 46–55 (1991)

15.96 K. K. Suresh, O. S. Deepa: Risk based Bayesian chain sampling plan, Far East J. Theor. Stat. **6**, 121–128 (2002)

15.97 E. G. Schilling: *Acceptance Sampling in Quality Control* (Marcel Dekker, New York 1982)

15.98 K. S. Stephens: *The Handbook of Applied Acceptance Sampling—Plans, Principles, and Procedures* (ASQ Quality, Milwaukee 2001)

16. Some Statistical Models for the Monitoring of High-Quality Processes

One important application of statistical models in industry is statistical process control. Many control charts have been developed and used in industry. They are easy to use, but have been developed based on statistical principles. However, for today's high-quality processes, traditional control-charting techniques are not applicable in many situations. Research has been going on in the last two decades and new methods have been proposed. This chapter summarizes some of these techniques.

High-quality processes are those with very low defect-occurrence rates. Control charts based on the cumulative count of conforming items are recommended for such processes. The use of such charts has opened up new frontiers in the research and applications of statistical control charts in general. In this chapter, several extended or modified statistical models are described. They are useful when the simple and basic geometric distribution is not appropriate or is insufficient.

In particular, we present some extended Poisson distribution models that can be used for count data with large numbers of zero counts. We also extend the chart to the case of general time-between-event monitoring; such an extension can be useful in service or reliability monitoring.

Traditionally, the exponential distribution is used for the modeling of time-between-events, although other distributions such as the Weibull or gamma distribution can also be used in this context.

Control charting is one of the most widely used statistical techniques in industry for process control and monitoring. It dates back to the 1920s when Walter Shewhart introduced the basic charting techniques in the United States [16.1]. Since then, it has been widely adopted worldwide, mainly in manufacturing and also in service industries. The simplicity of the application procedure allows a non-specialist user to observe the data and plot the control chart for simple decision making. At the same time, it provides sophisticated statistical interpretation in terms of false-alarm probability and average run length, among other important statistical properties associated with decision making based on sample information. The implementation of control charts had helped many companies to focus on important quality issues and problems such as those raised by out-of-control points on a control chart.

However, for high-quality or near-zero-defect processes, traditional Shewhart charts may not be suitable for process monitoring and decision making. This is especially the case for Shewhart attribute charts [16.2]. Many problems such as high false-alarm probability, inability to detect process improvement, unnecessary plotting of many zeros etc., have been identified by various researchers [16.3–6]. To resolve these problems, new models and monitoring techniques have been developed recently.

Traditional charts are all based on the principle of normal distribution and the upper control limit (UCL) and lower control limit (LCL) are routinely computed as the mean plus and minus three times the standard deviation. That is, if the plotted quantity Y has mean μ and standard deviation σ, the control limits are given by

$$\text{UCL} = \mu + 3\sigma \quad \textit{and} \quad \text{LCL} = \mu - 3\sigma \,. \tag{16.1}$$

Generally, when the distribution of Y is skewed, the probability of false alarm, i. e. the probability that a point indicating out-of-control when the process has actually not changed, is different from the nominal value of 0.0027 associated with a truly normal distribution. Note that for attribute charts, the plotted quantities usually follow a binomial or Poisson distribution, and this is far from the normal distribution unless the sample size is very large.

The purpose of this chapter is to review the important models and techniques that can be used to monitor high-quality processes. The procedure based on a general principle of the cumulative count of conforming items is first described; this is then extended to other distributions. The emphasis is on recent developments and also on practical methods that can be used by practitioners.

This chapter is organized as follows. First, the use of probability limits is described. Next, control charts based on monitoring of the cumulative count of conforming items and simple extensions are discussed. Control charts based on the zero-inflated Poisson distribution and generalize Poisson distribution are then presented. These charts are widely discussed in the literature and they are suitable for count or attribute data. For process monitoring, time-between-events monitoring is of growing importance, and we also provide a summary of methods that can be used to monitor process change based on time-between-events data. Typical models are the exponential, Weibull and gamma distribution.

16.1 Use of Exact Probability Limits

For high-quality processes it is important to use probability limits instead of traditional three-sigma limits. This is true when the quality characteristic that is being plotted follows a skewed distribution. For any plotted quality characteristic Y, the probability limits LCL_Y and UCL_Y can be derived as

$$P(X < \text{LCL}_Y) = P(X > \text{UCL}_Y) = \alpha/2 \,, \tag{16.2}$$

where α is the false-alarm probability, i. e., when the process is in control, the probability that the control chart raises an alarm. Assuming that the distribution $F(x)$ is known or has been estimated accurately from the data, the control limits can be computed.

Probability limits are very important for attribute charts as the quality characteristics are usually not normally distributed. If this is the case, the false-alarm probability could be much higher than the nominal value ($\alpha = 0.0027$ for traditional three-sigma limits). *Xie* and *Goh* [16.7] studied the exact probability limits calculated from the binomial distribution and the Poisson distribution applied for the *np* chart and the *c* chart.

For control-chart monitoring the number of nonconforming items in samples of size n, assuming that the process fraction nonconforming is p, the probability that there are exactly k nonconforming items in the sample is

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \,, \quad k = 0, 1, \ldots n \tag{16.3}$$

and the probability limits can be computed as

$$P(X \leqslant \text{LCL}) = \sum_{i=0}^{\text{LCL}} \binom{n}{i} p^i (1-p)^{n-i} = \frac{\alpha}{2} \tag{16.4}$$

and

$$P(X \leqslant \text{UCL}) = \sum_{i=0}^{\text{UCL}} \binom{n}{i} p^i (1-p)^{n-i} = 1 - \frac{\alpha}{2} \,. \tag{16.5}$$

As discussed, probability limits can be computed for any distributions, and should be used when the distribution is skewed. This will form the basis of the following discussion in this chapter. In some cases, although the solution is analytically intractable, they can be obtained with computer programs. It is advisable that probability limits be used unless the normality test indicates that deviation from normal distribution is not significant.

16.2 Control Charts Based on Cumulative Count of Conforming Items

High-quality processes are usually characterized by low defective rates. In a near-zero-defect manufacturing environment, items are commonly produced and inspected one-by-one, sometimes automatically. We can record and use the cumulative count of conforming items produced before a nonconforming item is detected. This technique has been intensively studied in recent years.

16.2.1 CCC Chart Based on Geometric Distribution

The idea of tracking cumulative count of conforming (CCC) items to detect the onset of assignable causes in an automated (high-quality) manufacturing environment was first introduced in [16.3]. *Goh* [16.4] further developed this idea into what is known as the CCC charting technique. Some related discussions and further studies can be found in [16.8–14], among others. *Xie* et al. [16.15] provided extensive coverage of this charting technique and further analysis of this procedure.

For a process with a defective rate of p, the cumulative count of conforming items before the appearance of a nonconforming item, Y, follows a geometric distribution. This is given by

$$P(Y = n) = (1 - p)^{n-1} p, \quad n = 1, 2, \ldots . \tag{16.6}$$

The cumulative probability function of count Y is given by

$$P(Y \leqslant n) = \sum_{i=1}^{n} (1 - p)^{i-1} p = 1 - (1 - p)^n . \tag{16.7}$$

Assuming that the acceptable false-alarm probability is α, the probability limits for the CCC chart are obtained as

$$\mathrm{UCL} = \ln(\alpha/2)/\ln(1 - p) \tag{16.8}$$

and

$$\mathrm{LCL} = \ln(1 - \alpha/2)/\ln(1 - p) \tag{16.9}$$

Usually the center line (CL) is computed as

$$\mathrm{CL} = \ln(1/2)/\ln(1 - p) . \tag{16.10}$$

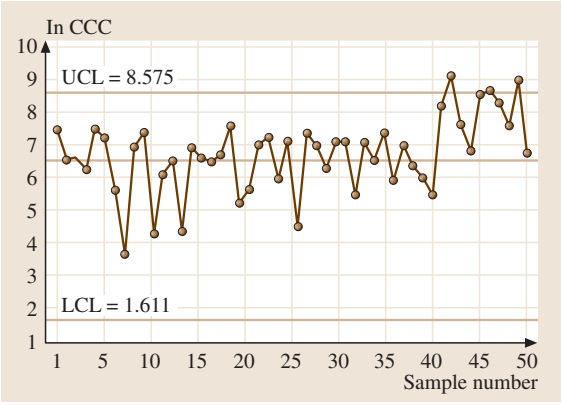


Fig. 16.1 A typical cumulative count of conforming (CCC) items chart

A typical CCC chart is shown in Fig. 16.1 The first 40 data points are simulated with $p = 0.001$ and the last one was simulated with $p = 0.0002$. The value of α is set to be 0.01 for the calculation of control limits. Note that we have also used a log scale for CCC.

Note that the decision rule is different from that of the traditional p or np chart. If a point is plotted above the UCL, the process is considered to have improved. When a point falls below the LCL, the process is judged to have deteriorated. An important advantage is that the CCC chart can detect not only the increase in the defective rate (process deterioration), but also the decrease in the defective rate (process improvement).

16.2.2 CCC-r Chart Based on Negative Binomial Distribution

A simple idea to generalize a CCC chart is to consider plotting of the cumulative count of items inspected until observing two nonconforming items. This was studied in [16.16] resulting in the CCC-2 control chart. This chart increases the sensitivity of the original CCC chart for the detection of small process shifts in p. The CCC-2 chart has smaller type II error, which is related to chart sensitivity, and steeper OC (Operating Characteristic) curves than the CCC chart with the same type I, error which is the false alarm probability.

A CCC-r chart [16.17,18] plots the cumulative count of items inspected until r nonconforming items are observed. This will further improve the sensitivity and detect small changes faster. However, it requires more

counts to be cumulated in order to generate an alarm signal. The CCC-r charting technique was also studied by *Lu* et al. [16.17].

Let Y be the cumulative count of items inspected until r nonconforming items have been observed. Let the probability of an item to be nonconforming be p. Then Y follows a negative binomial distribution given by

$$P(Y=n)=\binom{n-1}{r-1}p^r(1-p)^{n-r}\,,\quad n=r,r+1,\ldots\,. \tag{16.11}$$

The cumulative distribution function of count Y would be

$$\begin{aligned}F(n,r,p)&=\sum_{i=r}^{n}P(Y=i)\\&=\sum_{i=r}^{n}\binom{i-1}{r-1}p^r(1-p)^{i-r}\,.\end{aligned} \tag{16.12}$$

If the acceptable false-alarm probability is α, then the upper control limit and the lower control limit, UCL_r and LCL_r, respectively, of the CCC$-r$ chart can be obtained as the solution of the following equations:

$$\begin{aligned}F(\mathrm{UCL}_r,r,p)&=\sum_{i=r}^{\mathrm{UCL}_r}\binom{i-1}{r-1}p^r(1-p)^{i-r}\\&=1-\alpha/2\end{aligned} \tag{16.13}$$

and

$$F(\mathrm{LCL}_r,r,p)=\sum_{i=r}^{\mathrm{LCL}_r}\binom{i-1}{r-1}p^r(1-p)^{i-r}=\alpha/2\,. \tag{16.14}$$

Note that this chart is suitable for one-by-one inspection process and so no subjective sample size is needed. On the other hand, the selection of r is a subjective issue if the cost involved is not a consideration. As the value of r increases the sensitivity of the chart may increase, but the user probably needs to wait too long to plot a point. *Ohta* et al. [16.18] addressed this issue from an economic design perspective and proposed a simplified design method to select a suitable value of r based on the economic design method for control charts that monitor discrete quality characteristics.

16.3 Generalization of the *c*-Chart

The c-chart is based on monitoring of the number of defects in a sample. Traditionally, the number of defect in a sample follows the Poisson distribution. The control limits are computed as

$$\mathrm{UCL}=c+3\sqrt{c}\quad\text{and}\quad\mathrm{LCL}=c-3\sqrt{c}\,, \tag{16.15}$$

where c is the average number of defects in the sample and the LCL is set to be zero when the value computed with (16.15) is negative.

However, for high-quality processes, it has been shown that these limits may not be appropriate. Some extensions of this chart are described in this section.

16.3.1 Charts Based on the Zero-Inflated Poisson Distribution

In a near-zero-defect manufacturing environment, many samples will have no defects. However, for those containing defects, we have observed that there could be many defects in a sample and hence the data has an over-dispersion pattern relative to the Poisson distribution. To overcome this problem, a generalization of Poisson distribution was used in [16.6, 19].

This distribution is commonly called the zero-inflated Poisson distribution. Let Y be the number of defects in a sample; the probability mass function is given by

$$\begin{cases}P(Y=0)=(1-p)+p\,\mathrm{e}^{-\lambda}\\P(Y=d)=p\frac{\lambda^d\mathrm{e}^{-\lambda}}{d!}\quad d=1,2,\ldots\,.\end{cases} \tag{16.16}$$

This has an interesting interpretation. The process is basically zero-defect although it is affected by causes that lead to one or more defects. If the occurrence of these causes is p, and the severity is λ, then the number of defects in the sample will follow a zero-inflated Poisson distribution.

When the zero-inflated Poisson distribution provides a good fit to the data, two types of control charts can be applied. One is the exact probability limits control chart, and the other is the CCC chart. When implementing the exact probability limits chart, *Xie* and *Goh* [16.6] suggested that only the upper control limit n_u should be considered, since the process is in a near-zero-defect manufacturing environment and the probability of zero is usually very large. The upper control limit can be

determined by:

$$\sum_{d=n_u}^{\infty} p\frac{\lambda^d e^{-\lambda}}{d!} \le \alpha\,, \tag{16.17}$$

where α is the probability of the type I error. It should be noticed that n_u could easily be solved because it takes only discrete values.

Control charts based on the zero-inflated Poisson distribution commonly have better performance in the near-zero-defect manufacturing environment. However, the control procedure is more complicated than the traditional methods since more effort is required to test the suitability of this model with more parameters.

For the zero-inflation Poisson distribution we have that [16.20]

$$E(Y) = p\lambda \tag{16.18}$$

and

$$\mathrm{Var}(Y) = p\lambda + p\lambda(\mu - p\lambda)\,. \tag{16.19}$$

It should be pointed out that the zero-inflation Poisson model is very easy to use, as the mean and variance are of close form. For example, the moment estimates can be obtained straightforward. On the other hand, the maximum-likelihood estimates can also be obtained.

The maximum-likelihood estimates can be obtained by solving

$$\begin{cases} p = \dfrac{1 - n_0/n}{1 - \exp(-\lambda)} \\ \lambda = \bar{y}/p \end{cases}, \tag{16.20}$$

where $\bar{y} = \sum_{i=1}^{n} y_i/n$, [16.20].

When the count data can be fitted by a zero-inflation Poisson model, statistical process control procedures can be modified. Usually, the lower control limit for zero-inflation Poisson model will not exist, because the probability of zero is larger than the predetermined type I error level. This is common for the attribute control chart. In the following section, the upper control limit will be studied.

The upper control limit n_u for a control chart based on the number of nonconformities can be obtained as the smallest integer solution of the following equation:

$$P(n_u \text{ or more nonconformities in a sample}) \leqslant \alpha_L\,, \tag{16.21}$$

where α_L is the predetermined false-alarm probability for the upper control limit n_u.

Here our focus is on data modeling with appropriate distribution. It can be noted that the model contains two parameters. To be able to monitor the change in each parameter, a single chart may no be appropriate. *Xie* and *Goh* [16.6] developed a procedure for the monitoring of individual parameter. First, a CCC chart is used for data with zero count. Second, a *c*-chart is used for those with one or more non-zero count.

Note that a useful model should have practical interpretations. In this case, p is the occurrence probability of problem in the process, and λ measures the severity of the problem when it occurs. Hence it is a useful model and important to be able to monitor each of these parameters, so that any change from normal behavior can be identified.

Example 1

An example is used here for illustration [16.2]. The data set used in Table 16.1 is the read–write errors discovered in a computer hard disk in a manufacturing process.

For the data set in Table 16.1, it can be seen that it contains many samples with no nonconformities. From the data set, the maximum-likelihood estimates are $\hat{p} = 0.1346$ and $\hat{\mu} = 8.6413$. The overall zero-inflation

Table 16.1 A set of defect count data

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	6	0	9
11	0	1	2	0	0	0	0	0	0	0	0	3	3	0	0	5	0	15	6
0	0	0	4	2	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0
0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
0	0	1	0	0	0	0	0												

Poisson model for the data set is

$$f(y) = \begin{cases} 1 - 0.1346 + 0.1346\exp(-8.6413)\,, \\ \quad \text{if } y = 0\,, \\ 0.1346\frac{8.6413^y \exp(-8.6413)}{y!}\,, \\ \quad \text{if } y > 0\,. \end{cases} \tag{16.22}$$

For the data set in Table 16.1, it can be calculated that the upper control limit is 14 at an acceptable false-alarm rate of 0.01. This means that there should not be any alarm for values less than or equal to 14 when the underlying distribution model is a zero-inflated Poisson distribution.

16.3.2 Chart Based on the Generalized Poisson Distribution

The generalized Poisson distribution is another useful model that extends the traditional Poisson distribution, which only has one parameter. A two-parameter model is usually much more flexible and able to model different types of data sets. Since in the situation of over-dispersion or under-dispersion the Poisson distribution is no longer preferable as it must have equal mean and variance, the generalized Poisson distribution [16.21] can be used.

This distribution has two parameters (θ, λ) and the probability mass function is defined as

$$P_X(\theta, \lambda) = \frac{\theta(\theta + x\lambda)^{x-1}\,\mathrm{e}^{-\theta - x\lambda}}{x!}, \quad x = 0, 1, 2\ldots\,, \tag{16.23}$$

where $\lambda, \theta > 0$.

For the generalized Poisson distribution we have that [16.21]

$$E(X) = \theta(1-\lambda)^{-1} \tag{16.24}$$

and

$$\mathrm{Var}(X) = \theta(1-\lambda)^{-3}\,. \tag{16.25}$$

It should be pointed out that the generalized Poisson distribution model is very easy to use as both the mean and variance are of closed form. For example, the moment estimates can easily be calculated. On the other hand, the maximum-likelihood estimates can also be obtained straightforwardly. Consider a set of observations $\{X_1, X_2, \ldots, X_n\}$ with sample size n, the maximum-likelihood estimation $(\hat{\theta}, \hat{\lambda})$ can be obtained by solving

$$\begin{cases} \sum_{i=1}^{n} \frac{x_i(x_i - 1)}{\bar{x} + (x_i - \bar{x})\hat{\lambda}} - n\bar{x} = 0\,, \\ \hat{\theta} = \bar{x}(1 - \hat{\lambda})\,. \end{cases} \tag{16.26}$$

Here a similar approach as for the zero-inflated Poisson model can be used. One could also developed two charts for practical monitoring. One chart can be used to monitor the severity and another to monitor the dispersion or variability in terms of the occurrence of defects.

Example 2

The data in Table 16.1 can also be modeled with a generalized Poisson distribution. Based on the data, the maximum-likelihood estimates can be computed as $\hat{\theta} = 0.144297$ and $\hat{\lambda} = 0.875977$. The overall generalized Poisson distribution model for the data set is

$$f(x) = \frac{0.144297(0.144297 + 0.875977x)^{x-1}}{x!} \times \frac{\mathrm{e}^{-0.144297 - 0.875977x}}{x!}, \quad x = 0, 1, 2\ldots\,. \tag{16.27}$$

With this model, it can be calculated that the upper control limit is 26 at a false-alarm rate of 0.01. This means that there should not be any alarm for the values less than or equal to 26 when the underlying distribution model is the generalized Poisson distribution. It should be mentioned here that, for this data set, both models can fit the data well, and the traditional Poisson distribution is rejected by statistical tests.

16.4 Control Charts for the Monitoring of Time-Between-Events

Chan et al. [16.22] proposed a charting method called the cumulative quantity control chart (CQC chart). Suppose that defects in a process are observed according to a Poisson process with mean rate of occurrence equal to λ (>0). Then the number of units Q required to observe exactly one defect is an exponential random variable. The control chart for Q can be constructed to monitor possible shifts of λ in the process, which is the CQC chart.

The CQC chart has several advantages. It can be used for low-defective-rate processes as well as moderate-defective-rate processes. When the process defect rate

is low or moderate, the CQC chart does not have the shortcoming of showing up frequent false alarms. Furthermore, the CQC chart does not require rational grouping of samples. The data required is the time between defects or defective items. This type of data is commonly available in equipment and process monitoring for production and maintenance.

When process failures can be described by a Poisson process, the time between failures will be exponential and the same procedure can be used in reliability monitoring. Here we briefly describe the procedure for this type of monitoring. Since time is our preliminary concern, the control chart will be termed a t-chart in this paper. This is in line with the traditional c-chart or u-chart, to which our t-chart may be a more suitable alternative. In fact, the notation also makes it easier for the extension to be discussed later.

16.4.1 CQC Chart Based on the Exponential Distribution

The distribution function of the exponential distribution with parameter λ is given by

$$F(t;\lambda) = 1 - \mathrm{e}^{-\lambda t}, \quad t \geqslant 0\,. \tag{16.28}$$

The control limits for t-chart are defined in such a manner that the process is considered to be out of control when the time to observe exactly one failure is less than the lower control limit (LCL), T_{L}, or greater than the upper control limit (UCL), T_{U}. When the behavior of the process is normal, there is a chance for this to happen and it is commonly known as a false alarm. The traditional false-alarm probability is set to be 0.27%, although any other false-alarm probability can be used. The actual acceptable false-alarm probability should in fact depend on the actual product or process. Assuming an acceptable probability for false alarms of α, the control limits can be obtained from the exponential distribution as:

$$T_{\mathrm{L}} = \lambda^{-1} \ln \frac{1}{1-\alpha/2} \tag{16.29}$$

and

$$T_{\mathrm{U}} = \lambda^{-1} \ln \frac{2}{\alpha}\,. \tag{16.30}$$

The median of the distribution is the center line (CL), T_{C}, and it can be computed as

$$T_{\mathrm{C}} = \lambda^{-1} \ln 2 = 0.693\lambda^{-1}\,. \tag{16.31}$$

These control limits can then be utilized to monitor the failure times of components. After each failure the time can be plotted on the chart. If the plotted point falls between the calculated control limits, this indicates that the process is in the state of statistical control and no action is warranted. If the point falls above the upper control limit, this indicates that the process average, or the failure occurrence rate, may have decreased, resulting in an increase in the time between failures. This is an important indication of possible process improvement. If this happens the management should look for possible causes for this improvement and if the causes are discovered then action should be taken to maintain them. If the plotted point falls below the lower control limit, this indicates that the process average, or the failure occurrence rate, may have increased, resulting in a decrease in the failure time. This means that the process may have deteriorated and thus actions should be taken to identify and remove them.

In either case the people involved can know when the reliability of the system has changed and by a proper follow-up they can maintain and improve the reliability. Another advantage of using the control chart is that it informs the maintenance crew when to leave the process alone, thus saving time and resources.

16.4.2 Chart Based on the Weibull Distribution

It is well known that the lifetime distribution of many components follows a Weibull distribution [16.23]. Hence when monitoring reliability or equipment failure, this distribution has been shown to be very useful. The Weibull distribution function is given as

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\theta}\right)^{\beta}\right], \quad t \geq 0\,, \tag{16.32}$$

where $\theta > 0$ and $\beta > 0$ are the so called scale parameter and shape parameter, respectively.

The Weibull distribution is a generalization of exponential distribution, which is recovered for $\beta = 1$. Although the exponential distribution has been widely used for times-between-event, Weibull distribution is more suitable as it is more flexible and is able to deal with different types of aging phenomenon in reliability. Hence in reliability monitoring of equipment failures, the Weibull distribution is a good alternative.

A process can be monitored with a control chart and the time-between-events can be used. For the Weibull distribution, the control limits can be calculated as:

$$\mathrm{UCL} = \theta_0 \left[\ln\left(\frac{2}{\alpha}\right)\right]^{1/\beta_0} \tag{16.33}$$

and

$$\mathrm{LCL} = \theta_0 \left[\ln\left(\frac{2}{2-\alpha}\right)\right]^{1/\beta_0} , \qquad (16.34)$$

where α is the acceptable false-alarm probability, and β_0 and θ_0 are the in-control shape and scale parameter, respectively. Generally, the false-alarm probability is fixed at $\alpha = 0.0027$, which is equivalent to the three-sigma limits for an X-bar chart under the normal-distribution assumption.

The center line can be defined as

$$\mathrm{CL} = \theta_0 \, [\ln 2]^{1/\beta_0} . \qquad (16.35)$$

Xie et al. [16.24] carried out some detailed analysis of this procedure. Since this model has two parameters, a single chart may not be able to identify changes in a parameter. However, since in a reliability context, it is unlikely that the shape parameter will change and it is the scale parameter that could be affected by ageing or wear, a control chart as shown in Fig. 16.2 can be useful in reliability monitoring.

16.4.3 General *t*-Chart

In general, to model time-between-events, any distribution for positive random variables could be used. Which distribution is used should depend on the actual data, with the exponential, Weibull and Gamma being the most common distributions. However, these distributions are usually very skewed. The best approach is to use probability limits. It is also possible to use a transformation so that the data is transformed to near-normality, so that traditional chart for individual data can be used; such charting procedure is commonly available in statistical process control (SPC) software.

In general, if the variable Y follows the distribution $F(t)$, the probability limits can be computed as usual, that is:

$$F(\mathrm{LCL}_Y) = 1 - F(\mathrm{UCL}_Y) = \alpha/2 , \qquad (16.36)$$

where α is the fixed false-alarm rate. This is an approach that summarizes the specific cases described earlier. However, it is important to be able to identify the distribution to be used.

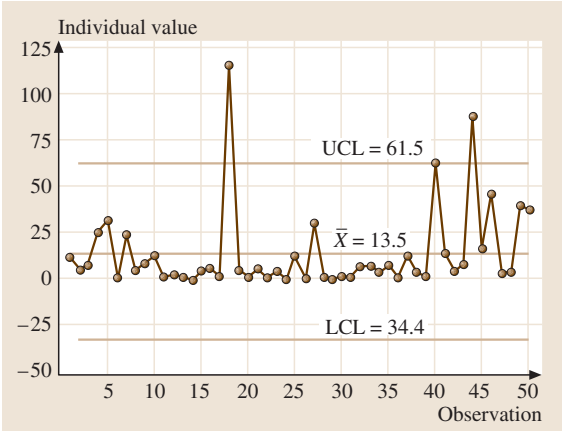


Fig. 16.2 A set of Weibull data and the plot

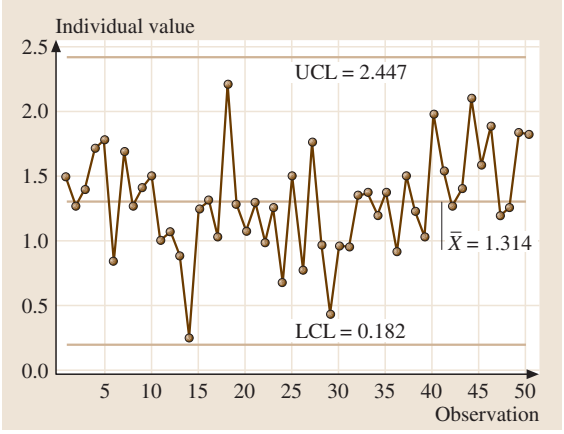


Fig. 16.3 The same data set as in Fig. 16.2 with the plot of the Box–Cox transformation

Furthermore, to make better use of the traditional monitoring approach, we could use a simple normality transformation. The most common ones are the Box–Cox transformation and the log or power transformations. They can be easily realized in software such as MINITAB. Figure 16.2 shows a chart for a Weibull-distributed process characteristic and Fig. 16.3 shows the individual chart with a Box–Cox transformation.

16.5 Discussion

In this chapter, some effective control-charting techniques are described. the statistical monitoring technique should be tailored to the specific distribution of the data that are collected from the process. Perfunctory use of

the traditional chart will not help much in today's manufacturing environment towards near-zero-defect process. For high-quality processes, it is more appropriate to monitor items inspected between two nonconforming items or the time between two events.

The focus in this article is to highlight some common statistical distributions for process monitoring. Several statistical models such as the geometric, negative binomial, zero-inflated Poisson, and generalized Poisson can be used for count-data monitoring in this context. The exponential, Weibull and Gamma distributions can be used to monitor time-between-events data, which is common in reliability or equipment failure monitoring. Other general distributions of time-between-events can also be used when appropriate. The approach is still simple: by computing the probability limits for a fixed false-alarm probability, any distribution can be used in a similar way. The simple procedure is summarized below:

Step 1. Study the process and identify the statistical distribution for the process characteristic;

Step 2. Collect data and estimate the parameters (and validate the model, if needed);

Step 3. Compute the probability limits or use an appropriate normality transformation with an individual chart;

Step 4. Identify any assignable cause and take appropriate action.

The distributions presented in this paper open the door to further implementation of statistical process control techniques in a near-zero-defect era. Several research issues remain. For example, the problem with correlated data and the estimation problem has to be studied. In a high-quality environment, failure or defect data is rare, and the estimation problem becomes serious. In the case of continuous production and measurement, data correlation also becomes an important issue. It is possible to extend the approach to consider the exponentially weighted moving-average (EWMA) or cumulative-sum (CUSUM) charts that are widely advocated by statisticians. A further area of importance is multivariate quality characteristics. However, a good balance between statistical performance and ease of implementation and understanding by practitioners is essential.

References

16.1 W. A. Shewhart: *Economic Control of Quality of Manufacturing Product* (Van Nostrand, New York 1931)

16.2 M. Xie, T. N. Goh: Some procedures for decision making in controlling high yield processes, Qual. Reliab. Eng. Int. **8**, 355–360 (1992)

16.3 T. W. Calvin: Quality control techniques for "zero-defects", IEEE Trans. Compon. Hybrids Manuf. Technol. **6**, 323–328 (1983)

16.4 T. N. Goh: A charting technique for control of low-nonconformity production, Int. J. Qual. Reliab. Man. **4**, 53–62 (1987)

16.5 T. N. Goh: Statistical monitoring, control of a low defect process, Qual. Reliab. Eng. Int. **7**, 497–483 (1991)

16.6 M. Xie, T. N. Goh: Improvement detection by control charts for high yield processes, Int. J. Qual. Reliab. Man. **10**, 24–31 (1993)

16.7 M. Xie, T. N. Goh: The use of probability limits for process control based on geometric distribution, Int. J. Qual. Reliab. Man. **14**, 64–73 (1997)

16.8 P. D. Bourke: Detecting shift in fraction nonconforming using run-length control chart with 100% inspection, J. Qual. Technol. **23**, 225–238 (1991)

16.9 F. C. Kaminsky, R. D. Benneyan, R. D. Davis, R. J. Burke: Statistical control charts based on ageometric distribution, J. Qual. Technol. **24**, 63–69 (1992)

16.10 E. A. Glushkovsky: On-line G-control chart for attribute data, Qual. Reliab. Eng. Int. **10**, 217–227 (1994)

16.11 C. P. Quesenberry: Geometric Q charts for high quality processes, J. Qual. Technol. **27**, 304–313 (1995)

16.12 W. Xie, M. Xie, T. N. Goh: Control charts for processes subject to random shocks, Qual. Reliab. Eng. Int. **11**, 355–360 (1995)

16.13 T. C. Chang, F. F. Gan: Charting techniques for monitoring a random shock process, Qual. Reliab. Eng. Int. **15**, 295–301 (1999)

16.14 Z. Wu, S. H. Yeo, H. T. Fan: A comparative study of the CRL-type control charts, Qual. Reliab. Eng. Int. **16**, 269–279 (2000)

16.15 M. Xie, T. N. Goh, P. Ranjan: Some effective control chart procedures for reliability monitoring, Reliab. Eng. Sys. Saf. **77**(2), 143–150 (2002)

16.16 L. Y. Chan, M. Xie, T. N. Goh: Two-stage control charts for high yield processes, Int. J. Reliab. Qual. Saf. Eng. **4**, 149–165 (1997)

16.17 M. Xie, X. S. Lu, T. N. Goh, L. Y. Chan: A quality monitoring, decision-making scheme for automated production processes, Int. J. Qual. Reliab. Man. **16**, 148–157 (1999)

16.18 H. Ohta, E. Kusukawa, A. Rahim: A CCC-*r* chart for high-yield processes, Qual. Reliab. Eng. Int. **17**, 439–446 (2001)
16.19 B. He, M. Xie, T. N. Goh, P. Ranjan: On the estimation error in zero-inflated Poisson model for process control, Int. J. Reliab. Qual. Saf. Eng. **10**, 159–169 (2003)
16.20 D. Bohning: Zero-inflated Poisson models, C.A.MAN: A tutorial collection of evidence, Biom. J. **40**, 833–843 (1998)
16.21 P. C. Consul: *Generalized Poisson Distributions: Properties and Applications* (Marcel Dekker, New York 1989)
16.22 L. Y. Chan, M. Xie, T. N. Goh: Cumulative quantity control charts for monitoring production processes, Int. J. Prod. Res. **38**(2), 397–408 (2000)
16.23 D. N. P. Murthy, M. Xie, R. Jiang: *Weibull Models* (Wiley, New York 2003)
16.24 M. Xie, T. N. Goh, V. Kuralmani: *Statistical Models and Control Charts for High Quality Processes* (Kluwer Academic, Boston 2002)

17. Monitoring Process Variability Using EWMA

During the last decade, the use of the exponentially weighted moving average (EWMA) statistic as a process-monitoring tool has become more and more popular in the statistical process-control field. If the properties and design strategies of the EWMA control chart for the mean have been thoroughly investigated, the use of the EWMA as a tool for monitoring process variability has received little attention in the literature. The goal of this chapter is to present some recent innovative EWMA-type control charts for the monitoring of process variability (i. e. the sample variance, sample standard-deviation and the range). In the first section of this chapter, the definition of an EWMA sequence and its main properties will be presented together with the commonly used procedures for the numerical computation of the average run length (ARL). The second section will be dedicated to the use of the EWMA as a monitoring tool for the process position, i. e. sample mean and sample median. In the third section, the use of the EWMA for monitoring the sample variance, sample standard deviation and the range will be presented, assuming a fixed sampling interval (FSI) strategy. Finally, in the fourth section of this chapter, the variable sampling interval adaptive version of the EWMA-S^2 and EWMA-R control charts will be presented.

During the last decade, the use of the exponentially weighted moving average (EWMA) statistic as a process monitoring tool has become increasingly popular in the field of statistical process control (SPC). If the properties and design strategies of the EWMA control chart for the mean (introduced by *Roberts* [17.1]) have been thoroughly investigated by *Robinson* and *Ho* [17.2], *Crowder* [17.3] [17.4], *Lucas* and *Saccucci* [17.5] and *Steiner* [17.6], the use of the EWMA as a tool for monitoring the process variability has received little attention in the literature. Some exceptions are the papers by *Wortham* and *Ringer* [17.7], *Sweet* [17.8], *Ng* and *Case* [17.9], *Crowder* and *Hamilton* [17.10], *Hamilton* and *Crowder* [17.11] and *MacGregor* and *Harris* [17.12], *Gan* [17.13], *Amin* et al. [17.14], *Lu* and *Reynolds* [17.15], *Acosta-Mejia* et al. [17.16] and *Castagliola* [17.17]. The goal of this chapter is to present some recent innovative EWMA-type control charts for the monitoring of process variability (i. e. the sample variance, sample standard deviation and the range). From an industrial perspective, the potential of EWMA charts is important. Since their pioneer applications, these charts have proved highly sensitivity in the detection of small shifts in the monitored process parameter, due to the structure of the plotted EWMA statistic, which takes into account the past history of the process at each

sampling time: this allowed them to be considered as valuable alternatives to the standard Shewhart charts, especially when the sample data needed to determine the EWMA statistic can be collected individually and evaluated automatically. As a consequence, the EWMAs have been implemented successfully on continuous processes such as those in chemical or food industries, where data involving operating variables such as temperatures, pressures, viscosity, etc. can be gathered and represented on the chart directly by the control system for the process. In the recent years, thanks to the development of simple quality-control software tools, that can be easily managed by workers and implemented on a common PC or notebook, use of EWMAs has systematically been extended to processes for manufacturing discrete parts; in this case, EWMAs that consider sample statistics like mean, median or sample variance are particularly well suited. Therefore, EWMA charts for monitoring process mean or dispersion have been successfully implemented in the semiconductor industry at the level of wafer fabrication; these processes are characterized by an extremely high level of precision in critical dimensions of parts and therefore there is the need of a statistical tool that is able to identify very small drifts in the process parameter to avoid the rejection of the product at the testing stage or, in the worst case, during the operating conditions, i.e., when the electronic device has been installed on highly expensive boards. Other applications of EWMAs to manufacturing processes involve the assembly operations in automotive industry, the technological processes involving the production of mechanical parts like CNC operations on machining centers, where process variability should be maintained as small as possible, and many others. Finally, it is important to note how EWMAs are also spreading in service control activities; an interesting example is represented by recent applications of EWMA charts to monitor healthcare outcomes such as the occurrence of infections or mortality rate after surgeries. Finally, EWMA charts can be adopted for any manufacturing process or service with a low effort and should always be preferred to Shewhart charts when there is the need to detect small shifts in the process parameters, as will be proven later in this chapter.

Therefore, in the second section of this chapter the definition of an EWMA sequence and its main properties will be presented together with the commonly used procedures for the numerical computation of the average run length (ARL). An important part of this section will focus on the numerical computation of the average run length (ARL). The third section will be dedicated to the use of the EWMA as a monitoring tool for the process position, i. e. sample mean (EWMA-$\bar{X}$) and sample median (EWMA-$\tilde{X}$). In the fourth section, the use of the EWMA for monitoring the sample variance (EWMA-S^2), sample standard deviation (EWMA-S) and the range (EWMA-R) will be presented, assuming a fixed sampling interval (FSI) strategy. In the fifth section the variable sampling interval adaptive version of the EWMA-S^2 and EWMA-R control charts will be presented.

The following notations are used – *ARL*: average run length; *ATS*: average time to signal, h_S, h_L: short and long sampling interval; K: width of the control limits; λ: EWMA smoothing parameter; *LCL*, *UCL*: lower and upper control limits; *LWL*, *UWL*: lower and upper warning limits; μ_0, σ_0: in-control mean and standard deviation; μ_1, σ_1: out-of-control mean and standard deviation; R, S, S^2: range, sample standard deviation and sample variance; τ: shift in the process position or dispersion; W: width of the warning limits; $\bar{X}$, $\tilde{X}$: sample mean and sample median.

17.1 Definition and Properties of EWMA Sequences

17.1.1 Definition

Let $T_1, \ldots, T_k, \ldots$ be a sequence of independently and identically distributed (i.i.d.) random variables and let $\lambda \in [0, 1]$ be a constant. From the sequence $T_1, \ldots, T_k, \ldots$ we define a new sequence $Y_1, \ldots, Y_k, \ldots$ using the following recurrence formula

$$Y_k = (1-\lambda)Y_{k-1} + \lambda T_k \,.$$

By decomposing Y_{k-1} in terms of Y_{k-2}, and Y_{k-2} in terms of Y_{k-3} and so on, it is straightforward to demonstrate that

$$Y_k = (1-\lambda)^k Y_0 + \lambda \sum_{j=0}^{k-1} (1-\lambda)^j T_{k-j} \,. \tag{17.1}$$

This formula clearly shows that Y_k is a linear combination of the initial random variable Y_0 weighted by a coefficient $(1-\lambda)^k$ and the random variables $T_1, \ldots, T_k$ weighted by the coefficients $\lambda(1-\lambda)^{k-1}, \lambda(1-\lambda)^{k-2}, \ldots, \lambda$. For this reason, the sequence $Y_1, \ldots, Y_k, \ldots$ is called an exponentially

weighted moving average (EWMA) sequence. If the random variables $T_1, \ldots, T_k, \ldots$ are, by definition, independent, the random variables $Y_1, \ldots, Y_k, \ldots$ are, by definition, not independent. We can notice that

- when $\lambda \to 0$ the sequence $Y_1, \ldots, Y_k, \ldots$ tends to be a smoothed version of the initial sequence $T_1, \ldots, T_k, \ldots$. When $\lambda = 0$ we have $Y_k = Y_{k-1} = \cdots = Y_0$.
- when $\lambda \to 1$ the sequence $Y_1, \ldots, Y_k, \ldots$ tends to be a copy of the initial sequence $T_1, \ldots, T_k, \ldots$ When $\lambda = 1$ we have $Y_k = T_k$ for $k \geq 1$.

17.1.2 Expectation and Variance of EWMA Sequences

Let $\mu_T = E(T_k)$ and $\sigma_T^2 = V(T_k)$ be the expectation and the variance of the random variables $T_1, \ldots, T_k, \ldots$ Using (17.1), we find that the expected value of the random variable Y_k is:

$$E(Y_k) = (1-\lambda)^k E(Y_0) + \lambda\mu_T \sum_{j=0}^{k-1} (1-\lambda)^j$$

or, equivalently, that

$$E(Y_k) = (1-\lambda)^k E(Y_0) + \mu_T[1-(1-\lambda)^k] .$$

Assuming $E(Y_0) = \mu_T$, for $k \geq 1$ it results that

$$E(Y_k) = \mu_T .$$

Because the random variables $T_1, T_2, \ldots$ are supposed to be independent, the variance $V(Y_k)$ of the random variable Y_k is

$$V(Y_k) = (1-\lambda)^{2k} V(Y_0) + \lambda^2\sigma_T^2 \sum_{j=0}^{k-1} (1-\lambda)^{2j} .$$

Replacing $\sum_{j=0}^{k-1}(1-\lambda)^{2j}$ by $[1-(1-\lambda)^{2k}]/[\lambda(2-\lambda)]$ gives

$$V(Y_k) = (1-\lambda)^{2k} V(Y_0) + \lambda^2\sigma_T^2 \left(\frac{1-(1-\lambda)^{2k}}{\lambda(2-\lambda)}\right) .$$

Finally, after some simplifications, we have

$$V(Y_k) = (1-\lambda)^{2k} V(Y_0) + \left(\frac{\lambda}{2-\lambda}\right)[1-(1-\lambda)^{2k}]\sigma_T^2 .$$

Two common assumptions can be made to determine the variance $V(Y_0)$ of the initial random variable Y_0:

- If we assume $V(Y_0) = 0$ (i. e., $Y_0 = \mu_T$ is a constant) then we have, for $k \geq 1$,

$$V(Y_k) = \left(\frac{\lambda}{2-\lambda}\right)[1-(1-\lambda)^{2k}]\sigma_T^2 .$$

- If we assume $V(Y_0) = \sigma_T^2$ then we have, for $k \geq 1$,

$$V(Y_k) = \left(\frac{\lambda + 2(1-\lambda)^{2k+1}}{2-\lambda}\right)\sigma_T^2 .$$

For either choice $V(Y_0) = 0$ or $V(Y_0) = \sigma_T^2$, the asymptotic variance $V_\infty(Y_k)$ of the random variable Y_k is

$$V_\infty(Y_k) = \lim_{k\to+\infty} V(Y_k) = \left(\frac{\lambda}{2-\lambda}\right)\sigma_T^2 .$$

17.1.3 The ARL for an EWMA Sequence

Let LCL and UCL be two constants satisfying $LCL < \mu_T < UCL$. Let $f_T(t)$ and $F_T(t)$ be the probability density function (p.d.f.) and the cumulative distribution function (c.d.f.) of the random variables $T_1, \ldots, T_k, \ldots$ Because the random variables $T_1, \ldots, T_k, \ldots$ are assumed to be independent, the average run length ARL_T for the sequence $T_1, \ldots, T_k, \ldots$ is given by

$$ARL_T = \frac{1}{F_T(LCL) + 1 - F_T(UCL)} . \qquad (17.2)$$

Let $ARL_Y(y)$ be the average run length of the EWMA sequence $Y_1, \ldots, Y_k, \ldots$ assuming $Y_0 = y$ and let $ARL_Y = ARL_Y(\mu_T)$. The fact that the random variables $Y_1, \ldots, Y_k, \ldots$ are not independent prevents the use of (17.2) for computing $ARL_Y(y)$. There are two main approaches for computing ARL_Y.

The first approach is based on the fact that $ARL_Y(y)$ must satisfy the following equation

$$ARL_Y(y) = 1 + \int_{LCL}^{UCL} \frac{1}{\lambda} \times f_T\left(\frac{z-(1-\lambda)y}{\lambda}\right) ARL_Y(z)\,\mathrm{d}z .$$

This equation is a Fredholm equation of the second kind, i. e.

$$f(y) = g(y) + \int_{z_1}^{z_n} h(y,z) f(z)\,\mathrm{d}z ,$$

where $h(y,z)$ and $g(y)$ are two known functions and where $f(z)$ is an unknown function that satisfies the equation above. In our case, we have $g(y)=1$, $h(y,z)=f_T[z-(1-\lambda)y]/\lambda/\lambda$ and $f(y)=ARL_Y(y)$. The numerical evaluation of a Fredholm equation (see *Press* et al. [17.18]) of the second kind consists of approximating the integral operand by a weighted sum

$$f(y) \simeq g(y) + \sum_{i=1}^{n} w_i h(y, z_i) f(z_i)\,, \tag{17.3}$$

where $z_1, z_2, \ldots, z_n$ and $w_1, w_2, \ldots, w_n$ are, respectively, the abscissas and weights of a quadrature method on $[z_1, z_n]$ such as, for instance, the n-point Gauss–Legendre quadrature. If we apply (17.3) for $y = z_1, z_2, \ldots, z_n$, we have

$$\begin{aligned}
f_1 &\simeq g_1 + w_1 h_{1,1} f_1 + w_2 h_{1,2} f_2 + \cdots + w_n h_{1,n} f_n\,,\\
f_2 &\simeq g_2 + w_1 h_{2,1} f_1 + w_2 h_{2,2} f_2 + \cdots + w_n h_{2,n} f_n\,,\\
&\vdots\ \vdots\ \vdots\\
f_n &\simeq g_n + w_1 h_{n,1} f_1 + w_2 h_{n,2} f_2 + \cdots + w_n h_{n,n} f_n\,,
\end{aligned}$$

where $f_i = f(z_i)$, $g_i = g(z_i)$ and $h_{i,j} = h(z_i, z_j)$. This set of equations can be rewritten in a matrix form

$$\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \simeq \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} + \begin{pmatrix} w_1 h_{1,1} & w_2 h_{1,2} & \cdots & w_n h_{1,n} \\ w_1 h_{2,1} & w_2 h_{2,2} & \cdots & w_n h_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_1 h_{n,1} & w_2 h_{n,2} & \cdots & w_n h_{n,n} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

or in a more compact way as

$$\boldsymbol{f} \simeq \boldsymbol{g} + \mathbf{H}\boldsymbol{f}\,.$$

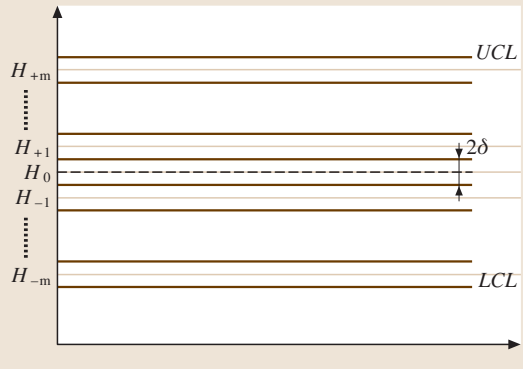


Fig. 17.1 Interval between *LCL* and *UCL* divided into $p = 2m+1$ subintervals of width 2δ

Solving the equation above for $\boldsymbol{f}$, we obtain $(\mathbf{I}-\mathbf{H})\boldsymbol{f} \simeq \boldsymbol{g}$, and finally

$$\boldsymbol{f} \simeq (\mathbf{I}-\mathbf{H})^{-1}\boldsymbol{g}\,.$$

The second approach is based on the flexible and relatively easy to use Markov-chain approach, originally proposed by *Brook* and *Evans* [17.19]. This procedure involves dividing the interval between *LCL* and *UCL* (Fig. 17.1) into $p = 2m+1$ subintervals of width 2δ, where $\delta = (UCL - LCL)/(2p)$. When the number of subintervals p is sufficiently large, the finite approach provides an effective method that allows ARL_Y to be accurately evaluated. The statistic $Y_k = (1-\lambda)Y_{k-1} + \lambda T_k$ is said to be in *transient state* j at time k if $H_j - \delta < Y_k < H_j + \delta$ for $j = -m, \ldots, -1, 0, +1, \ldots, m$, where H_j represents the midpoint of the j-th subinterval. The statistic Y_k is in the *absorbing state* if $Y_k \notin [LCL, UCL]$. An approximation for ARL_Y is given by

$$ARL_Y \simeq \boldsymbol{d}^{\mathrm{T}}\mathbf{Q}\boldsymbol{g}\,,$$

where $\boldsymbol{d}$ is the $(p,1)$ initial probability vector, $\mathbf{Q} = (\mathbf{I} - \mathbf{P})^{-1}$ is the fundamental (p,p) matrix, $\mathbf{P}$ is the (p,p) transition-probabilities matrix and $\boldsymbol{g} = \boldsymbol{1}$ is a $(p,1)$ vector of 1s. The initial probability vector $\boldsymbol{d}$ contains the probabilities that the statistic Y_k starts in a given state. This vector is such that, for $j = -m, \ldots, -1, 0, +1, \ldots, m$,

$$d_j = \begin{cases} 1 & \text{if } H_j - \delta < Y_0 < H_j + \delta \\ 0 & \text{otherwise}\,. \end{cases}$$

This vector contains a single entry equal to 1, whereas its $2m$ remaining elements are all equal to 0. The transition-probability matrix $\mathbf{P}$ contains the one-step transition probabilities. The generic element $p_{i,j}$ of $\mathbf{P}$ represents the probability that the statistic Y_k goes from state i to state j in one step. In order to approximate this probability, it is assumed that the statistic Y_k is equal to H_j whenever it is in state j, i. e.

$$p_{i,j} \simeq P\left(\frac{H_j - \delta - (1-\lambda)H_i}{\lambda} < T_k < \frac{H_j + \delta - (1-\lambda)H_i}{\lambda}\right).$$

This probability can be rewritten

$$p_{i,j} \simeq F_T\left(\frac{H_j + \delta - (1-\lambda)H_i}{\lambda}\right) - F_T\left(\frac{H_j - \delta - (1-\lambda)H_i}{\lambda}\right).$$

17.2 EWMA Control Charts for Process Position

17.2.1 EWMA-$\bar{X}$ Control Chart

Let $X_{k,1}, \ldots, X_{k,n}$ be a sample of n independent normal (μ_0, σ_0) random variables, where μ_0 is the nominal process mean, σ_0 is the nominal process standard deviation and k is the subgroup number. Let $\bar{X}_k$ be the sample mean of subgroup k, i. e.,

$$\bar{X}_k = \frac{1}{n}\sum_{j=1}^{n} X_{k,j} \,.$$

Traditional Shewhart control charts $(\bar{X}, R)$ or $(\bar{X}, S)$ directly monitor the sample mean $\bar{X}_k$, in contrast to EWMA-$\bar{X}$ control charts, which monitor the statistic $Y_k = (1-\lambda)Y_{k-1} + \lambda\bar{X}_k$, i. e., $T_k = \bar{X}_k$. This implies that $\mu_T = \mu_0$ and $\sigma_T = \sigma_0/\sqrt{n}$ and, consequently, the (fixed) control limits of the EWMA-$\bar{X}$ control chart (introduced by *Roberts* [17.1]) are

$$LCL = \mu_0 - K\sqrt{\frac{\lambda}{2-\lambda}}\frac{\sigma_0}{\sqrt{n}} \,,$$

$$UCL = \mu_0 + K\sqrt{\frac{\lambda}{2-\lambda}}\frac{\sigma_0}{\sqrt{n}} \,,$$

where K is a positive constant.

Example 17.1: Figure 17.2 reports a simulation of 200 data obtained from a manufacturing process: the 150 first data plotted in Fig. 17.2 consist of $m = 30$ subgroups of $n = 5$ observations randomly generated from a normal $(\mu_0 = 20, \sigma_0 = 0.1)$ distribution; the remaining 50 data are collected within 10 subgroups of $n = 5$ observations randomly generated from a normal (20.05, 0.1) distribution: that is, the process position was shifted up by half the nominal standard deviation. The process mean and standard deviation were estimated by considering the subgroups $1, \ldots, 30$, corresponding to the in-control condition: $\hat{\mu}_0 = 19.99$, $\hat{\sigma}_0 = 0.099$. Assuming $K = 3$, the control limits of the EWMA chart are, respectively, equal to:

- if $\lambda = 0.1$, $LCL = 19.959$ and $UCL = 20.020$,
- if $\lambda = 0.3$, $LCL = 19.934$ and $UCL = 20.046$.

In Fig. 17.3, we plot the EWMA-$\bar{X}$ control charts for the cases $\lambda = 0.1$ and $\lambda = 0.3$. We can see that these control charts detect an out-of-control signal at the 34-th subgroup (in the case $\lambda = 0.1$) and at the 33-rd subgroup (in the case $\lambda = 0.3$), point-

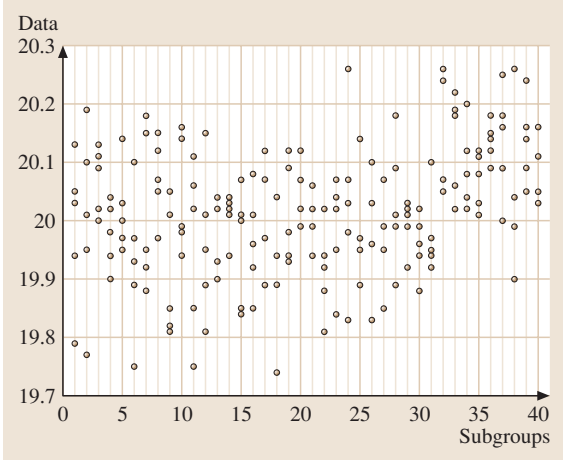


Fig. 17.2 Data with a half-standard-deviation shift in the process mean/median

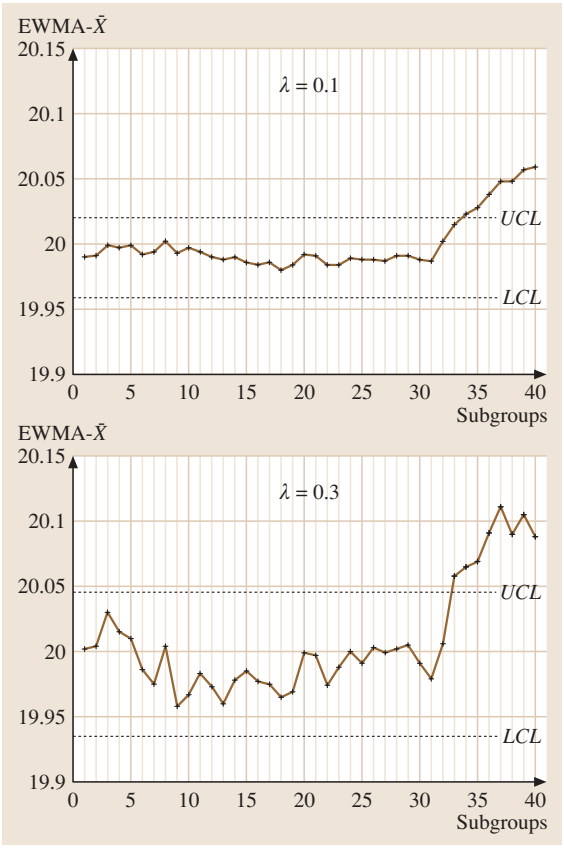


Fig. 17.3 EWMA-$\bar{X}$ control charts corresponding to the data in Fig. 17.2 for $\lambda = 0.1$ and $\lambda = 0.3$

ing out that an increasing of the process position occurred.

17.2.2 EWMA-$\tilde{X}$ Control Chart

Let $X_{k,(1)}, \ldots, X_{k,(n)}$ be the ordered sample corresponding to $X_{k,1}, \ldots, X_{k,n}$ and let $\tilde{X}_k$ be the sample median of subgroup k, i. e.

$$\tilde{X}_k = \begin{cases} X_{k,[(n+1)/2]} & \text{if } n \text{ is odd} \\ \dfrac{X_{k,(n/2)} + X_{k,(n/2+1)}}{2} & \text{if } n \text{ is even}. \end{cases}$$

The EWMA-$\tilde{X}$ control chart is a natural extension of the EWMA-$\bar{X}$ control chart investigated by *Castagliola* [17.20] where the monitored statistic is $Y_k = (1-\lambda)Y_{k-1} + \lambda\tilde{X}_k$, i. e., $T_k = \tilde{X}_k$. The (fixed) control limits of the EWMA-$\tilde{X}$ control chart are

$$LCL = \mu_0 - K\sqrt{\frac{\lambda}{2-\lambda}}\sigma(\tilde{X}),$$

$$UCL = \mu_0 + K\sqrt{\frac{\lambda}{2-\lambda}}\sigma(\tilde{X}),$$

where $\sigma(\tilde{X})$ is the standard deviation of the sample median $\tilde{X}$. It is straightforward to show that $\sigma(\tilde{X}) = \sigma_0 \times \sigma(\tilde{Z})$ where $\sigma(\tilde{Z})$ is the standard deviation of the normal $(0, 1)$ sample median. The values of $\sigma(\tilde{Z})$ are tabulated in Table 17.1 for $n \in \{3, 5, \ldots, 25\}$, but they can also be computed, when n is odd, (see *Castagliola* [17.21]), using the following approximation

$$\sigma(\tilde{Z}) \simeq \sqrt{\frac{\pi}{2(n+2)} + \frac{\pi^2}{4(n+2)^2} + \frac{\pi^2(\frac{13}{24}\pi - 1)}{2(n+2)^3}}.$$

Table 17.1 Standard-deviation $\sigma(\tilde{Z})$ of the normal $(0, 1)$ sample median, for $n \in \{3, 5, \ldots, 25\}$

n	$\sigma(\tilde{Z})$
3	0.6698
5	0.5356
7	0.4587
9	0.4076
11	0.3704
13	0.3418
15	0.3189
17	0.3001
19	0.2842
21	0.2707
23	0.2589
25	0.2485

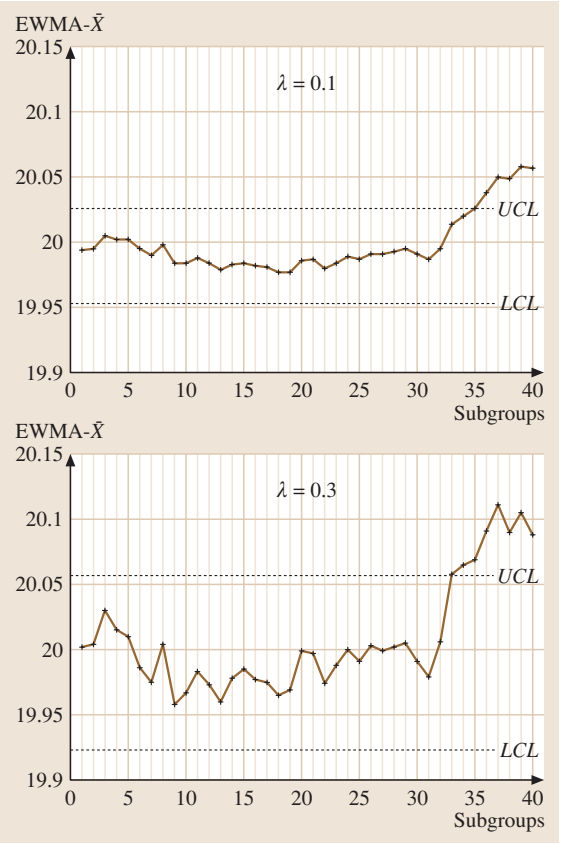


Fig. 17.4 EWMA-$\tilde{X}$ control charts corresponding to the data in Fig. 17.2 for $\lambda = 0.1$ and $\lambda = 0.3$

Example 17.2: using the same data (Fig. 17.2) as in the previous example, we computed the control limits of the EWMA-$\tilde{X}$ control chart ($K = 3$ assumed)

- if $\lambda = 0.1$, $LCL = 19.953$ and $UCL = 20.026$,
- if $\lambda = 0.3$, $LCL = 19.923$ and $UCL = 20.057$.

In Fig. 17.4 we plot the EWMA-$\tilde{X}$ control charts for the cases $\lambda = 0.1$ and $\lambda = 0.3$. Similarly to the EWMA-$\bar{X}$ control charts, we can see that these control charts detect an out-of-control signal at the 36-th subgroup (in the case $\lambda = 0.1$) and at the 33-rd subgroup (in the case $\lambda = 0.3$), pointing out again that an increasing of the process position occurred.

17.2.3 ARL Optimization for the EWMA-$\bar{X}$ and EWMA-$\tilde{X}$ Control Charts

Let $\tau = |\mu_1 - \mu_0|/\sigma_0$ be the usual variable reflecting the shift in the process position, where μ_1 is the new out-

Table 17.2 Optimal couples (λ^*, K^*) and optimal ARL^* of the EWMA-$\bar{X}$ (*half top*) and EWMA-$\tilde{X}$ (*half bottom*) control charts, for $\tau \in \{0.1, 0.2, \ldots, 2\}$, $n \in \{1, 3, 5, 7, 9\}$ and $ARL_0 = 370.4$

EWMA-$\bar{X}$																				
τ	$n=1$				$n=3$				$n=5$				$n=7$				$n=9$			
	λ^*	K^*	ARL^*	$\bar{X}$	λ^*	K^*	ARL^*	$\bar{X}$	λ^*	K^*	ARL^*	$\bar{X}$	λ^*	K^*	ARL^*	$\bar{X}$	λ^*	K^*	ARL^*	$\bar{X}$
0.1	0.01	1.824	183.0	352.9	0.01	1.824	103.5	322.1	0.02	2.139	77.1	295.8	0.02	2.139	62.2	273.0	0.02	2.139	53.1	253.1
0.2	0.01	1.824	87.8	308.4	0.03	2.305	43.9	227.7	0.04	2.414	31.0	177.7	0.05	2.492	24.5	143.9	0.07	2.601	20.4	119.7
0.3	0.02	2.139	53.1	253.1	0.05	2.492	25.1	147.5	0.08	2.640	17.4	99.5	0.10	2.703	13.5	72.7	0.12	2.749	11.2	55.8
0.4	0.04	2.414	36.2	200.1	0.08	2.640	16.6	94.0	0.12	2.749	11.3	56.6	0.15	2.801	8.8	38.3	0.18	2.840	7.3	27.8
0.5	0.05	2.492	26.5	155.2	0.11	2.727	11.9	60.7	0.17	2.828	8.1	33.4	0.21	2.869	6.3	21.4	0.25	2.898	5.2	15.0
0.6	0.07	2.601	20.4	119.7	0.15	2.801	9.0	40.0	0.22	2.877	6.1	20.6	0.28	2.916	4.7	12.7	0.33	2.938	3.9	8.7
0.7	0.08	2.640	16.3	92.3	0.19	2.850	7.2	27.1	0.27	2.910	4.9	13.2	0.34	2.941	3.8	8.0	0.41	2.962	3.1	5.4
0.8	0.10	2.703	13.4	71.6	0.23	2.885	5.8	18.8	0.33	2.938	4.0	8.9	0.42	2.964	3.1	5.3	0.51	2.979	2.5	3.6
0.9	0.12	2.749	11.2	55.8	0.27	2.910	4.9	13.4	0.39	2.957	3.3	6.2	0.50	2.978	2.6	3.7	0.61	2.989	2.1	2.6
1.0	0.14	2.786	9.6	43.9	0.31	2.930	4.2	9.8	0.46	2.972	2.8	4.5	0.59	2.988	2.2	2.8	0.70	2.995	1.8	2.0
1.1	0.16	2.815	8.3	34.8	0.36	2.948	3.6	7.3	0.53	2.982	2.4	3.4	0.67	2.993	1.9	2.2	0.77	2.997	1.5	1.6
1.2	0.18	2.840	7.3	27.8	0.41	2.962	3.2	5.6	0.60	2.989	2.1	2.7	0.74	2.996	1.6	1.8	0.83	2.999	1.3	1.4
1.3	0.21	2.869	6.4	22.4	0.46	2.972	2.8	4.4	0.67	2.993	1.9	2.2	0.80	2.998	1.4	1.5	0.88	2.999	1.2	1.2
1.4	0.23	2.885	5.7	18.2	0.52	2.981	2.5	3.5	0.73	2.996	1.7	1.8	0.85	2.999	1.3	1.3	0.92	3.000	1.1	1.1
1.5	0.25	2.898	5.2	15.0	0.58	2.987	2.2	2.9	0.79	2.998	1.5	1.6	0.89	2.999	1.2	1.2	0.95	3.000	1.1	1.1
1.6	0.28	2.916	4.7	12.4	0.63	2.991	2.0	2.4	0.83	2.999	1.4	1.4	0.93	3.000	1.1	1.1	0.97	3.000	1.0	1.0
1.7	0.30	2.925	4.3	10.3	0.68	2.994	1.8	2.1	0.87	2.999	1.3	1.3	0.95	3.000	1.1	1.1	0.99	3.000	1.0	1.0
1.8	0.33	2.938	3.9	8.7	0.73	2.996	1.7	1.8	0.90	3.000	1.2	1.2	0.97	3.000	1.0	1.0	0.99	3.000	1.0	1.0
1.9	0.36	2.948	3.6	7.4	0.77	2.997	1.5	1.6	0.93	3.000	1.1	1.1	0.98	3.000	1.0	1.0	1.00	3.000	1.0	1.0
2.0	0.38	2.954	3.3	6.3	0.81	2.998	1.4	1.5	0.95	3.000	1.1	1.1	0.99	3.000	1.0	1.0	1.00	3.000	1.0	1.0
0.1	0.01	1.824	183.0	352.9	0.01	2.116	122.1	335.2	0.01	2.184	95.0	317.7	0.01	2.214	79.5	301.4	0.02	2.615	68.5	286.3
0.2	0.01	1.824	87.8	308.4	0.02	2.481	53.4	258.3	0.03	2.760	39.7	218.3	0.04	2.930	32.1	187.2	0.05	3.047	27.2	162.6
0.3	0.02	2.139	53.1	253.1	0.04	2.800	30.9	182.6	0.06	3.057	22.6	137.4	0.07	3.157	18.0	107.5	0.09	3.270	15.2	86.8
0.4	0.04	2.414	36.2	200.1	0.07	3.018	20.6	125.2	0.09	3.204	14.8	85.4	0.11	3.312	11.8	62.2	0.14	3.409	9.9	47.5
0.5	0.05	2.492	26.5	155.2	0.09	3.104	14.9	85.7	0.13	3.318	10.6	53.9	0.16	3.420	8.4	37.1	0.19	3.488	7.0	27.2
0.6	0.07	2.601	20.4	119.7	0.12	3.192	11.3	59.2	0.17	3.391	8.1	34.9	0.21	3.487	6.4	23.0	0.24	3.541	5.3	16.4
0.7	0.08	2.640	16.3	92.3	0.15	3.254	9.0	41.6	0.21	3.441	6.4	23.2	0.26	3.532	5.1	14.8	0.30	3.583	4.2	10.4
0.8	0.10	2.703	13.4	71.6	0.18	3.299	7.3	29.7	0.25	3.478	5.2	15.9	0.31	3.564	4.1	9.9	0.37	3.617	3.5	6.9
0.9	0.12	2.749	11.2	55.8	0.21	3.334	6.1	21.6	0.29	3.506	4.4	11.2	0.37	3.592	3.5	6.9	0.44	3.640	2.9	4.8
1.0	0.14	2.786	9.6	43.9	0.25	3.370	5.2	16.0	0.34	3.533	3.7	8.1	0.43	3.612	3.0	5.0	0.52	3.657	2.5	3.5
1.1	0.16	2.815	8.3	34.8	0.28	3.391	4.5	12.0	0.39	3.553	3.2	6.0	0.50	3.629	2.6	3.8	0.60	3.669	2.1	2.7
1.2	0.18	2.840	7.3	27.8	0.32	3.414	4.0	9.2	0.45	3.572	2.8	4.6	0.57	3.641	2.2	2.9	0.67	3.677	1.8	2.1
1.3	0.21	2.869	6.4	22.4	0.36	3.432	3.5	7.2	0.51	3.585	2.5	3.6	0.64	3.650	2.0	2.3	0.74	3.682	1.6	1.8
1.4	0.23	2.885	5.7	18.2	0.40	3.447	3.2	5.7	0.57	3.595	2.2	2.9	0.70	3.655	1.7	2.0	0.80	3.685	1.4	1.5
1.5	0.25	2.898	5.2	15.0	0.45	3.461	2.8	4.6	0.63	3.603	2.0	2.4	0.76	3.659	1.6	1.7	0.84	3.687	1.3	1.3
1.6	0.28	2.916	4.7	12.4	0.49	3.470	2.6	3.8	0.68	3.608	1.8	2.1	0.81	3.662	1.4	1.5	0.88	3.688	1.2	1.2
1.7	0.30	2.925	4.3	10.3	0.54	3.480	2.3	3.2	0.73	3.612	1.6	1.8	0.85	3.663	1.3	1.3	0.92	3.688	1.1	1.1
1.8	0.33	2.938	3.9	8.7	0.59	3.487	2.1	2.7	0.78	3.615	1.5	1.6	0.88	3.664	1.2	1.2	0.94	3.689	1.1	1.1
1.9	0.36	2.948	3.6	7.4	0.64	3.493	2.0	2.4	0.82	3.617	1.4	1.4	0.91	3.665	1.1	1.2	0.96	3.689	1.1	1.1
2.0	0.38	2.954	3.3	6.3	0.68	3.497	1.8	2.1	0.85	3.618	1.3	1.3	0.94	3.665	1.1	1.1	0.98	3.689	1.0	1.0

of-control process position. The *ARL* of the EWMA-$\bar{X}$ and EWMA-$\tilde{X}$ control charts can be computed using one of the methods presented in Sect. 17.2.3. The p.d.f and c.d.f. $f_T(t)$ and $F_T(t)$, required for the computation of ARL_Y, are

- for the EWMA-$\bar{X}$ control chart

$$f_T(t) = \sqrt{n}\phi[(t-\tau)\sqrt{n}]$$
$$F_T(t) = \Phi[(t-\tau)\sqrt{n}]$$

  where $\phi(x)$ and $\Phi(x)$ are the p.d.f. and the c.d.f. of the normal (0, 1) distribution.
- for the EWMA-$\tilde{X}$ control chart

$$f_T(t) = \phi(t-\tau) f_\beta\left[\Phi(t-\tau)|\frac{n+1}{2}, \frac{n+1}{2}\right],$$
$$F_T(t) = F_\beta\left[\Phi(t-\tau)|\frac{n+1}{2}, \frac{n+1}{2}\right],$$

  where $f_\beta(x|a,b)$ and $F_\beta(x|a,b)$ are the p.d.f. and the c.d.f. of the (a,b) beta distribution.

The quality practitioner should be interested in determining the optimal couples (λ^*, K^*) that allow one to achieve:

- $ARL_Y = ARL_0$, where ARL_0 is the in-control *ARL*, corresponding to the process functioning at nominal position $\mu = \mu_0$, that is to $\tau = 0$;
- A minimum value for the out-of-control *ARL*, $ARL_Y = ARL_Y^*$, valid when $\tau > 0$.

Each couple (λ^*, K^*) is then optimally designed for detecting a particular shift τ. We chose to take $ARL_0 = 370.4$ for the in-control *ARL* (corresponding to the classical 3σ Shewhart control limits). In order to compute the couples (λ^*, K^*), we adopted the following approach

1. For every $\lambda \in \{0.01, 0.02, \ldots, 1\}$ and for $\tau = 0$, we computed (using a basic Newton-type algorithm) the corresponding value K such that $ARL_Y = ARL_0$. At the end of this step, we have a set of pairs $\{(0.01, K_{0.01}), (0.02, K_{0.02}), \ldots, (1, K_1)\}$ candidating for the second step.
2. For every shift $\tau \in \{0.1, 0.2, \ldots, 2\}$, and for every pair $\{(0.01, K_{0.01}), (0.02, K_{0.02}), \ldots, (1, K_1)\}$ we computed ARL_Y and chose the pair (λ^*, K^*) that gave the minimum $ARL_Y = ARL_Y^*$.

The optimal couples (λ^*, K^*) and the corresponding minimal ARL^* are shown in Table 17.2, for both the EWMA-$\bar{X}$ and EWMA-$\tilde{X}$ control charts. In Table 17.2, we also added, for comparison purpose, the *ARL* of the Shewhart $\bar{X}$ and $\tilde{X}$ control charts. For example, the optimal couple (λ^*, K^*) ensuring the smallest *ARL* for a shift $\tau = 0.5$ and $n = 7$ are $(0.21, 2.869)$ for the EWMA-$\bar{X}$ control chart and $(0.16, 3.420)$ for the EWMA-$\tilde{X}$ control chart. In that case, the minimal *ARL* is $ARL^* = 6.3$ for the EWMA-$\bar{X}$ control chart ($ARL = 21.4$ for the $\bar{X}$ control chart) and $ARL^* = 8.4$ for the EWMA-$\tilde{X}$ control chart ($ARL = 37.1$ for the $\tilde{X}$ control chart). Table 17.2 clearly demonstrates that, in terms of *ARL*, the EWMA-$\bar{X}$ control chart is more efficient than the EWMA-$\tilde{X}$ control chart and both the EWMA-$\bar{X}$ and EWMA-$\tilde{X}$ control charts are more efficient, for small and medium shift, than the traditional $\bar{X}$ and $\tilde{X}$ control charts.

17.3 EWMA Control Charts for Process Dispersion

17.3.1 EWMA-S^2 Control Chart

Let S_k^2 be the sample variance of subgroup k, i. e.,

$$S_k^2 = \frac{1}{n-1}\sum_{j=1}^{n}(X_{k,j} - \bar{X}_k)^2,$$

where $\bar{X}_k$ is the sample mean of subgroup k. In order to monitor the process variance, *Crowder* and *Hamilton* [17.10], following a recommendation by *Box* [17.22], suggested the application of the classical EWMA approach to the logarithm of the successive sample variances, i. e. $T_k = \ln S_k^2$. The main motivation for this approach is that $T_k = \ln S_k^2$, which has a log-gamma distribution (*Johnson* et al. [17.23]), tends to be more normally distributed than the sample variance S_k^2. A more recent idea developed by *Castagliola* [17.24] was to apply a three-parameter $(a_{S^2}, b_{S^2}, c_{S^2})$ logarithmic transformation to S_k^2, i. e. $T_k = a_{S^2} + b_{S^2}\ln(S_k^2 + c_{S^2})$, with $c_{S^2} > 0$ (in order to avoid problems with the logarithmic transformation). The main expectation of this approach is that, if the parameters a_{S^2}, b_{S^2} and c_{S^2} are judiciously selected, then this transformation may result in better normality of T_k than the approach of *Crowder* and *Hamilton* [17.10]. Trying to make T_k more normally distributed is related to making the distribu-

tion of T_k more symmetric. If the value of the expectation $E(T_k)$ and the standard deviation $\sigma(T_k)$ of T_k corresponding to the parameters a_{S^2}, b_{S^2} and c_{S^2} are known, then the (fixed) control limits of the EWMA-S^2 control chart are

$$LCL = E(T_k) - K\sqrt{\frac{\lambda}{2-\lambda}}\sigma(T_k)\,, \tag{17.4}$$

$$UCL = E(T_k) + K\sqrt{\frac{\lambda}{2-\lambda}}\sigma(T_k)\,. \tag{17.5}$$

The control limits given above correspond to a two-sided EWMA control chart, but one-sided EWMA control limits can also be considered. The approach suggested by *Castagliola* [17.24] needs to define a_{S^2}, b_{S^2}, c_{S^2}, $E(T_k)$, $\sigma(T_k)$ and Y_0. Let S^2 be the sample variance of n independent normal $(\mu_0, 1)$ random variables. The p.d.f. and the c.d.f of S^2 are defined for $s \geq 0$ and are equal to

$$f_{S^2}(s) = f_\gamma\left(s|\frac{n-1}{2}, \frac{2}{n-1}\right)\,,$$

$$F_{S^2}(s) = F_\gamma\left(s|\frac{n-1}{2}, \frac{2}{n-1}\right)\,,$$

where $f_\gamma(x|u, v)$ and $F_\gamma(x|u, v)$ are the p.d.f. and c.d.f. of the gamma (u, v) distribution

$$f_\gamma(x|u, v) = \begin{cases} 0 & (x \leq 0) \\ \dfrac{x^{u-1}\exp(-x/v)}{v^u\,\Gamma(u)} & (x > 0)\,. \end{cases}$$

Consequently, the expectation $E(S^2)$, the variance $V(S^2)$, and the skewness coefficient $\gamma_3(S^2)$ of S^2 are equal to

$$E(S^2) = 1\,,$$

$$V(S^2) = \frac{2}{n-1}\,,$$

$$\gamma_3(S^2) = \sqrt{\frac{8}{n-1}}\,.$$

In the approach developed by *Castagliola* [17.24], three parameters $A_{S^2}(n)$, $B_{S^2}(n)$, and $C_{S^2}(n)$, depending only on n, must be computed such that $T = A_{S^2}(n) + B_{S^2}(n)\ln[S^2 + C_{S^2}(n)]$ is approximately a normal $(0, 1)$ random variable. Remembering that S^2 has a gamma distribution, i. e. a unimodal skewed distribution, Castagliola suggested to find the three parameters log-normal distribution (another skewed distribution), defined for $x \geq -C_{S^2}(n)$,

$$f_L(x) = \frac{B_{S^2}(n)}{x}\phi\left[A_{S^2}(n) + B_{S^2}(n)\ln[x + C_{S^2}(n)]\right]\,,$$

which is the closest to the distribution of S^2, according to a criterion based on the first three moments $E(S^2)$, $V(S^2)$, and $\gamma_3(S^2)$ of S^2. If S^2 would have a log-normal distribution, then $T = A_{S^2}(n) + B_{S^2}(n)\ln[S^2 + C_{S^2}(n)]$ would have exactly a normal $(0, 1)$ distribution. By looking for a log-normal distribution which fits approximately the (gamma) distribution of S^2, we expect that T will have an approximate normal $(0, 1)$ distribution. What Castagliola suggested is to find parameters $A_{S^2}(n)$, $B_{S^2}(n)$, and $C_{S^2}(n)$ such that the log-normal distribution $f_L(x)$ fits the first three moments $E(S^2)$, $V(S^2)$, and $\gamma_3(S^2)$ of S^2. This can be achieved (*Stuart* and *Ord* [17.25]) by firstly computing

$$w = \left[\sqrt{\gamma_3^2(S^2)/4 + 1} + \gamma_3(S^2)/2\right]^{1/3} - \left[\sqrt{\gamma_3^2(S^2)/4 + 1} - \gamma_3(S^2)/2\right]^{1/3}$$

and then

$$B_{S^2}(n) = \frac{1}{\sqrt{\ln(w^2+1)}}\,,$$

$$A_{S^2}(n) = \frac{B_{S^2}(n)}{2}\ln\left(\frac{w^2(w^2+1)}{V(S^2)}\right)\,,$$

$$C_{S^2}(n) = \frac{\sqrt{V(S^2)}}{w} - E(S^2)\,.$$

The value of the constants $A_{S^2}(n)$, $B_{S^2}(n)$ and $C_{S^2}(n)$ can be found in Table 17.3 for $n \in \{3, \ldots, 15\}$. For the random variable $S_k^2 = \sigma_0^2 S^2$ [corresponding to the sample variance of n independent normal (μ_0, σ_0) random variables], it is straightforward to see that

$$\begin{aligned} T &= A_{S^2}(n) + B_{S^2}(n)\ln\left[S^2 + C_{S^2}(n)\right] \\ &= A_{S^2}(n) + B_{S^2}(n)\ln\left[S_k^2/\sigma_0^2 + C_{S^2}(n)\right] \\ &= A_{S^2}(n) - 2B_{S^2}(n)\ln(\sigma_0) + B_{S^2}(n)\ln\left[S_k^2 + C_{S^2}(n)\sigma_0^2\right]\,. \end{aligned}$$

Consequently, if the parameters a_{S^2}, b_{S^2} and c_{S^2} are defined such that

$$b_{S^2} = B_{S^2}(n)\,,$$

$$c_{S^2} = C_{S^2}(n)\sigma_0^2\,,$$

$$a_{S^2} = A_{S^2}(n) - 2B_{S^2}(n)\ln(\sigma_0)\,,$$

then $T = a_{S^2} + b_{S^2}\ln(S_k^2 + c_{S^2}) = T_k$. This ensures that $T_k = a_{S^2} + b_{S^2}\ln(S_k^2 + c_{S^2})$ will also be approximately a normal $(0, 1)$ random variable. Because $T_k = T$, the distribution $f_T(t)$ of T_k depends only on n. This distribution is defined for $t \geq A_{S^2}(n) + B_{S^2}(n)\ln[C_{S^2}(n)]$ and

Table 17.3 Constants $A_{S^2}(n)$, $B_{S^2}(n)$, $C_{S^2}(n)$, Y_0, $E(T_k)$, $\sigma(T_k)$, $\gamma_3(T_k)$ and $\gamma_4(T_k)$ for the EWMA-S^2 control chart, for $n \in \{3, \ldots, 15\}$

EWMA-S^2								
n	$A_{S^2}(n)$	$B_{S^2}(n)$	$C_{S^2}(n)$	Y_0	$E(T_k)$	$\sigma(T_k)$	$\gamma_3(T_k)$	$\gamma_4(T_k)$
3	−0.6627	1.8136	0.6777	0.276	0.024 72	0.9165	0.5572	−0.3206
4	−0.7882	2.1089	0.6261	0.237	0.012 66	0.9502	0.3752	−0.3947
5	−0.8969	2.3647	0.5979	0.211	0.007 48	0.9670	0.2746	−0.3803
6	−0.9940	2.5941	0.5801	0.193	0.004 85	0.9765	0.2119	−0.3478
7	−1.0827	2.8042	0.5678	0.178	0.003 35	0.9825	0.1697	−0.3142
8	−1.1647	2.9992	0.5588	0.167	0.002 43	0.9864	0.1398	−0.2837
9	−1.2413	3.1820	0.5519	0.157	0.001 82	0.9892	0.1176	−0.2572
10	−1.3135	3.3548	0.5465	0.149	0.001 41	0.9912	0.1007	−0.2344
11	−1.3820	3.5189	0.5421	0.142	0.001 12	0.9927	0.0874	−0.2147
12	−1.4473	3.6757	0.5384	0.136	0.000 90	0.9938	0.0768	−0.1978
13	−1.5097	3.8260	0.5354	0.131	0.000 74	0.9947	0.0681	−0.1831
14	−1.5697	3.9705	0.5327	0.126	0.000 62	0.9955	0.0610	−0.1703
15	−1.6275	4.1100	0.5305	0.122	0.000 52	0.9960	0.0550	−0.1591

Table 17.4 Optimal couples (λ^*, K^*) and optimal ARL^* for the EWMA-S^2 control chart, for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{3, 5, 7, 9\}$ and $ARL_0 = 370.4$

EWMA-S^2																
τ	$n=3$				$n=5$				$n=7$				$n=9$			
	λ^*	K^*	ARL^*	S^2	λ^*	K^*	ARL^*	S^2	λ^*	K^*	ARL^*	S^2	λ^*	K^*	ARL^*	S^2
0.60	0.10	2.690	17.8	267.0	0.17	2.782	9.0	102.2	0.25	2.836	6.2	45.6	0.31	2.863	4.7	23.6
0.70	0.08	2.644	28.5	363.1	0.12	2.724	14.6	184.5	0.17	2.791	10.0	102.5	0.20	2.823	7.7	62.3
0.80	0.05	2.547	54.3	467.0	0.08	2.635	28.2	308.2	0.10	2.689	19.5	211.8	0.12	2.733	15.1	152.5
0.90	0.05	2.547	155.8	512.1	0.05	2.515	81.0	445.8	0.05	2.505	56.8	384.1	0.06	2.556	44.7	333.1
0.95	0.05	2.547	325.7	463.7	0.05	2.515	202.9	451.0	0.05	2.505	150.6	433.1	0.05	2.501	120.9	414.7
1.05	0.05	2.547	153.6	268.8	0.05	2.515	121.4	253.5	0.05	2.505	99.3	242.1	0.05	2.501	84.0	232.3
1.10	0.05	2.547	64.9	186.4	0.05	2.515	44.8	159.6	0.05	2.505	34.9	140.8	0.05	2.501	29.0	126.2
1.20	0.05	2.547	21.6	90.1	0.05	2.515	15.3	64.5	0.05	2.505	12.4	50.0	0.05	2.501	10.7	40.5
1.30	0.05	2.547	11.6	48.0	0.05	2.515	8.8	30.5	0.05	2.505	7.4	22.0	0.05	2.501	6.6	16.9
1.40	0.05	2.547	7.8	28.5	0.05	2.515	6.2	16.8	0.05	2.505	5.3	11.7	0.30	2.861	4.7	8.8
1.50	0.05	2.547	5.9	18.6	0.05	2.515	4.8	10.5	0.42	2.848	4.2	7.2	0.46	2.864	3.4	5.4
1.60	0.05	2.547	4.8	13.1	0.05	2.515	4.0	7.2	0.49	2.839	3.2	4.9	0.54	2.854	2.7	3.7
1.70	0.05	2.547	4.0	9.8	0.05	2.515	3.5	5.3	0.57	2.827	2.6	3.6	0.65	2.836	2.2	2.8
1.80	0.05	2.547	3.5	7.7	0.55	2.830	2.9	4.2	0.66	2.811	2.2	2.9	0.73	2.822	1.9	2.2
1.90	0.05	2.547	3.1	6.2	0.55	2.830	2.5	3.4	0.66	2.811	2.0	2.4	0.77	2.815	1.6	1.9
2.00	0.05	2.547	2.9	5.2	0.63	2.829	2.3	2.9	0.74	2.799	1.8	2.0	0.78	2.813	1.5	1.6

is equal to

$$f_T(t) = \frac{1}{B_{S^2}(n)} \exp\left(\frac{t - A_{S^2}(n)}{B_{S^2}(n)}\right) \times f_{S^2}\left[\exp\left(\frac{t - A_{S^2}(n)}{B_{S^2}(n)}\right) - C_{S^2}(n) | n\right].$$

The fact that the distribution $f_T(t)$ of T_k depends only on n is important since it allows the calculation of the values of $E(T_k)$ and $\sigma(T_k)$ independently of the value of σ_0. The computation of $E(T_k)$ and $\sigma(T_k)$ has been achieved by numerical quadrature for $n \in \{3, \ldots, 15\}$, and the results are also shown in Table 17.3. Due to the fact that $E(T_k)$ approximates 0, whatever the sample size n, setting $E(T_k) = 0$ does not introduce significant errors into the statistical model. Finally, it seems logical to define the first value $Y_0 = E[a_{S^2} + b_{S^2} \ln(S^2 + c_{S^2})]$.

Using the first-order expansion of the expectation, we deduce $Y_0 \simeq a_{S^2} + b_{S^2} \ln(\sigma_0^2 + c_{S^2})$. Then by replacing a_{S^2}, b_{S^2} and c_{S^2} with $A_{S^2}(n) - 2B_{S^2}(n)\ln(\sigma_0)$, $B_{S^2}(n)$ and $C_{S^2}(n)\sigma_0^2$, we have

$$Y_0 \simeq A_{S^2}(n) - 2B_{S^2}(n)\ln(\sigma_0) + B_{S^2}(n)\ln[\sigma_0^2 + C_{S^2}(n)\sigma_0^2] .$$

After simplifications, we obtain:

$$Y_0 \simeq A_{S^2}(n) + B_{S^2}(n)\ln[1 + C_{S^2}(n)] .$$

As can be noticed, this value depends only on n and not on σ_0. The values for Y_0 are given in Table 17.3. One can note that these values are also close to 0 and can be replaced by 0 in practice with little practical effect.

Example 17.3: The goal of this example is to show how the EWMA-S^2 control chart behaves in the case of an increase and a decrease in the nominal process variability. The first 100 data points plotted in Fig. 17.5 (top and bottom) consist of 20 identical subgroups of $n = 5$ observations randomly generated from a normal (20, 0.1) distribution (corresponding to an in-control process), while the last 50 data points of Fig. 17.5 (top) consist of 10 subgroups of $n = 5$ observations randomly generated from a normal (20, 0.2) distribution (the nominal process standard deviation σ_0 has increased by a factor of 2), and the last 50 data points of Fig. 17.5 (bottom) consist of 10 subgroups of $n = 5$ observations randomly generated from a normal (20, 0.05) distribution (the nominal process standard deviation σ_0 has decreased by a factor of 2). The corresponding 30 sample variances are plotted in Fig. 17.6, for the two cases (increasing and

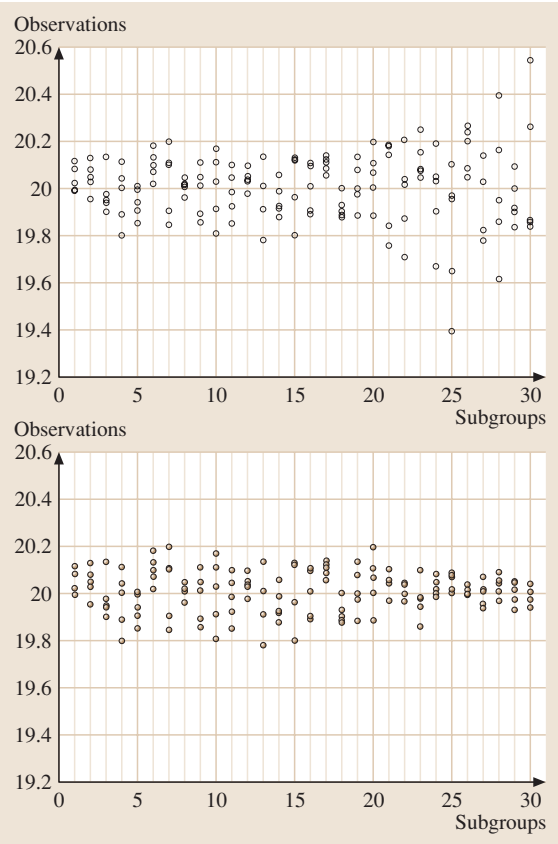


Fig. 17.5 Data with an increasing variance (*top*), and with a decreasing variance (*bottom*)

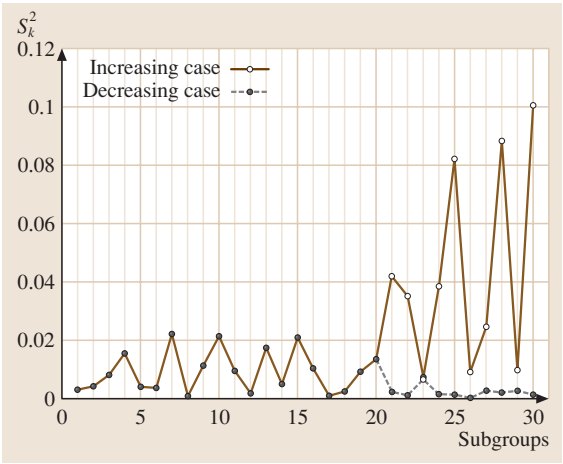


Fig. 17.6 Sample variances S_k^2 corresponding to the data of Fig. 17.5

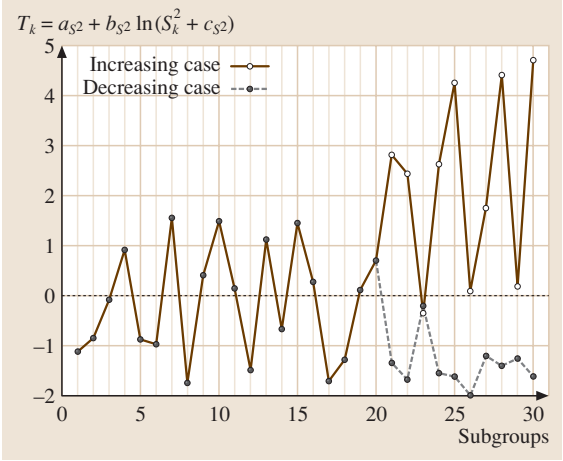


Fig. 17.7 Transformed sample variances $T_k = a_{S^2} + b_{S^2} \ln(S_k^2 + c_{S^2})$ corresponding to the data of Fig. 17.5

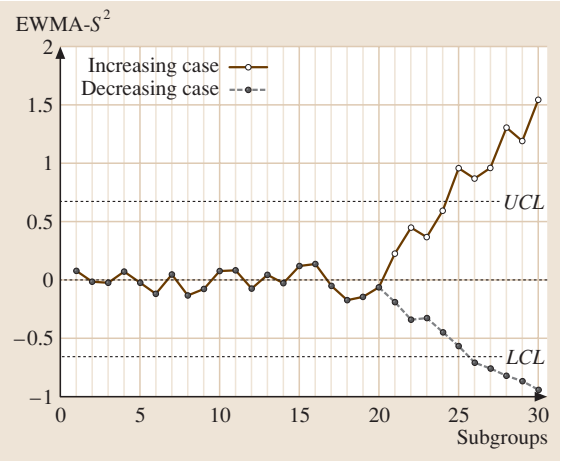


Fig. 17.8 EWMA-S^2 control chart ($\lambda = 0.1$, $K = 3$) corresponding to the data of Fig. 17.5

decreasing). At this step, the asymmetry between increasing and decreasing sample variances is particularly noticeable. If $n = 5$ and $\sigma_0 = 0.1$, then $b_{S^2} = 2.3647$, $c_{S^2} = 0.5979 \times 0.1^2 = 0.005\,979$ and $a_{S^2} = -0.8969 - 2 \times 2.3647 \times \ln(0.1) = 9.9929$. The 30 transformed sample variances are plotted in Fig. 17.7 and the EWMA-S^2 sequence along with the EWMA-S^2 control limits $LCL = 0.0075 - 3\sqrt{0.1/1.9} \times 0.967 = -0.658$ and $UCL = 0.0075 + 3\sqrt{0.1/1.9} \times 0.967 = 0.673$ ($\lambda = 0.1$ and $K = 3$) are plotted in Fig. 17.8. The EWMA-S^2 control chart clearly detects an out-of-control signal at the 25-th subgroup (in the increasing case) and at the 26-th subgroup (in the decreasing case), pointing out that an increase/decrease of the process variability occurred.

The distribution $f_T(t)$ of $T_k = a_{S^2} + b_{S^2} \ln(S_k^2 + c_{S^2})$ (plain line) for $n \in \{3, 5, 7, 9\}$, and the normal (0, 1) distribution (dotted line) are plotted in Fig. 17.9. The distribution of T_k is defined for $t \geq A_{S^2}(n) + B_{S^2}(n) \ln[C_{S^2}(n)]$, while the normal (0, 1) distribution is defined on $]-\infty, +\infty[$. This difference is particularly important for $n = 3$. It is clear that when n increases, the distribution of T_k becomes closer to the normal (0, 1) distribution, and the lower bound $A_{S^2}(n) + B_{S^2}(n) \ln[C_{S^2}(n)] \to -\infty$. By numerical quadrature, the skewness coefficient $\gamma_3(T_k) = \mu_3(T_k)/V^{3/2}(T_k)$ and the kurtosis coefficient $\gamma_4(T_k) = \mu_4(T_k)/V^2(T_k) - 3$ of T_k have been computed for $n = 3, \ldots, 15$. The results are shown in Table 17.3. As expected, when n increases, T_k becomes more normally distributed, i. e., $\gamma_3(T_k) \to 0$ and $\gamma_4(T_k) \to 0$.

Let σ_1 be the new out-of-control process standard deviation and let $\tau = \sigma_1/\sigma_0$ be the variable reflecting the shift in the process variability. Let $S_k'^2 = \sigma_1^2 S^2$ be the sample variance of n independent normal (μ_0, σ_1) random variables (i. e. the sample variance after a shift τ), and let $T_k' = a_{S^2} + b_{S^2} \ln(S_k'^2 + c_{S^2})$. If a_{S^2}, b_{S^2}, c_{S^2} and $S_k'^2$ are respectively replaced by $A_{S^2}(n) - 2B_{S^2}(n) \ln(\sigma_0)$, $B_{S^2}(n)$, $\sigma_0 C_{S^2}(n)$ and $\sigma_1^2 S^2$, then

$$\begin{aligned} T_k' &= A_{S^2}(n) - 2B_{S^2}(n) \ln(\sigma_0) \\ &\quad + B_{S^2}(n) \ln\left[\sigma_1^2 S^2 + \sigma_0^2 C_{S^2}(n)\right] \\ &= A_{S^2}(n) - 2B_{S^2}(n) \ln(\sigma_0) \\ &\quad + B_{S^2}(n) \ln\left\{\sigma_0^2 [\tau^2 S^2 + C_{S^2}(n)]\right\} \\ &= A_{S^2}(n) + B_{S^2}(n) \ln[\tau^2 S^2 + C_{S^2}(n)]\,. \end{aligned}$$

This result clearly shows that the distribution $f_{T'}(t)$ of T_k' is equal to the distribution of the transformed random variable $\tau^2 S^2$. Consequently the p.d.f and the c.d.f of T_k' are

$$\begin{aligned} f_{T'}(t) &= \frac{1}{\tau^2 B_{S^2}(n)} \exp\left(\frac{t - A_{S^2}(n)}{B_{S^2}(n)}\right) \\ &\quad \times f_{S^2}\left\{\frac{1}{\tau^2}\left[\exp\left(\frac{t - A_{S^2}(n)}{B_{S^2}(n)}\right) - C_{S^2}(n)\right] | n\right\}, \\ F_{T'}(t) &= F_{S^2}\left\{\frac{1}{\tau^2}\left[\exp\left(\frac{t - A_{S^2}(n)}{B_{S^2}(n)}\right) - C_{S^2}(n)\right] | n\right\}. \end{aligned}$$

The ARL of the EWMA-S^2 control chart can be computed using one of the methods presented in Sect. 17.1.3. Like for the EWMA-$\bar{X}$ and EWMA-$\tilde{X}$ control charts, it is sometimes interesting for the quality practitioner to know the optimal couples (λ^*, K^*) that give the same in-control $ARL_Y = ARL_0$ (i. e. the ARL when the process is functioning at the nominal variability $\sigma = \sigma_0$ or equivalently $\tau = 1$) and then find, for a specified value of the shift τ, the unique couple (λ^*, K^*) which yields the smallest possible out-of-control $ARL_Y = ARL^*$. In order to compute the couples (λ^*, K^*) for the EWMA-S^2 control chart, the same approach as in the EWMA-$\bar{X}$ and EWMA-$\tilde{X}$ control charts was adopted:

1. For every $\lambda \in \{0.05, 0.06, \ldots, 1\}$ and for $\tau = 1$, we computed the corresponding value K such that $ARL_Y = ARL_0$. At the end of this step, we have a set of pairs $\{(0.05, K_{0.05}), (0.06, K_{0.06}), \ldots, (1, K_1)\}$ candidating for the second step.
2. For every shift $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, and for every pair $\{(0.05, K_{0.05}), (0.06, K_{0.06}), \ldots, (1, K_1)\}$ we computed ARL_Y and chose the pair (λ^*, K^*) which gave the minimum $ARL_Y = ARL_Y^*$.

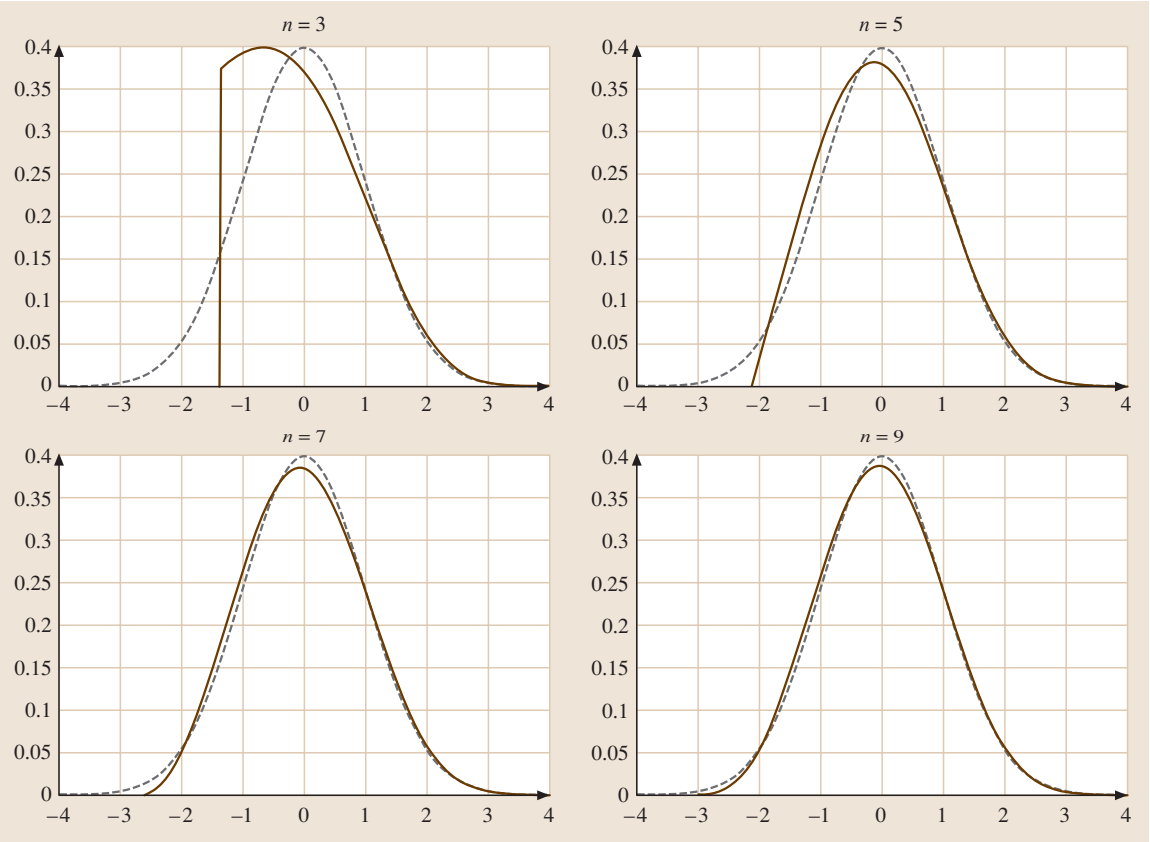


Fig. 17.9 Distribution $f_T(t)$ of $T_k = a_{S^2} + b_{S^2} \ln(S_k^2 + c_{S^2})$ (*plain line*) compared with the normal $(0, 1)$ distribution (*dotted line*), for $n \in \{3, 5, 7, 9\}$

The optimal couples (λ^*, K^*) for the EWMA-S^2 control chart and the corresponding minimal ARL^* are shown in Table 17.4. In Table 17.4, we also added, for comparison purpose, the ARL of the classic S^2 control chart. For example, the optimal couple (λ^*, K^*) ensuring the smallest ARL for a shift $\tau = 1.2$ and $n = 7$ is $(0.05, 2.505)$ and the corresponding minimal ARL is $ARL^* = 12.4$, while for the classic S^2 control chart, we have $ARL = 50.0$. This is just an example of the superiority of the EWMA-S^2 control chart (with optimized parameters) over the classical S^2 control chart when the shift τ is small.

17.3.2 EWMA-S Control Chart

The EWMA-S control chart proposed by *Castagliola* [17.26] is a natural extension of the EWMA-S^2 control chart where a three-parameter (a_S, b_S, c_S) logarithmic transformation is applied to the sample standard deviation S_k, [i. e. $T_k = a_S + b_S \ln(S_k + c_S)$], instead of the sample variance S_k^2. The control limits of the EWMA-S control chart are given by (17.4) and (17.5) but with different values for $E(T_k)$ and $\sigma(T_k)$. The p.d.f. and the c.d.f. of S are defined for $s \geq 0$ and are equal to

$$f_S(s) = 2s f_\gamma\left(s^2 | \frac{n-1}{2}, \frac{2}{n-1}\right),$$
$$F_S(s) = F_\gamma\left(s^2 | \frac{n-1}{2}, \frac{2}{n-1}\right),$$

and the mean $E(S)$, the variance $V(S)$, and the skewness coefficient $\gamma_3(S)$ of S are equal to

$$E(S) = K_S(n, 1),$$
$$V(S) = 1 - K_S^2(n, 1),$$

Table 17.5 Constants $A_S(n)$, $B_S(n)$, $C_S(n)$, Y_0, $E(T_k)$, $\sigma(T_k)$, $\gamma_3(T_k)$ and $\gamma_4(T_k)$ for the EWMA-S control chart, for $n \in \{3, \ldots, 15\}$

EWMA-S								
n	$A_S(n)$	$B_S(n)$	$C_S(n)$	Y_0	$E(T_k)$	$\sigma(T_k)$	$\gamma_3(T_k)$	$\gamma_4(T_k)$
3	−3.8134	4.8729	1.3474	0.1026	0.000 92	0.9917	0.1361	−0.4489
4	−5.4669	6.2696	1.5009	0.0797	0.000 30	0.9965	0.0741	−0.3221
5	−6.8941	7.4727	1.5984	0.0669	0.000 14	0.9981	0.0472	−0.2454
6	−8.1528	8.5370	1.6650	0.0586	0.000 07	0.9988	0.0331	−0.1964
7	−9.2839	9.4980	1.7131	0.0526	0.000 04	0.9992	0.0248	−0.1628
8	−10.3158	10.3789	1.7493	0.0482	0.000 03	0.9994	0.0194	−0.1387
9	−11.2684	11.1958	1.7776	0.0447	0.000 02	0.9996	0.0157	−0.1206
10	−12.1562	11.9605	1.8002	0.0418	0.000 01	0.9997	0.0130	−0.1066
11	−12.9901	12.6813	1.8186	0.0394	0.000 01	0.9997	0.0110	−0.0955
12	−13.7783	13.3650	1.8340	0.0374	0.000 01	0.9998	0.0095	−0.0864
13	−14.5272	14.0164	1.8469	0.0357	0.000 01	0.9998	0.0082	−0.0789
14	−15.2418	14.6397	1.8581	0.0342	0.000 01	0.9998	0.0073	−0.0725
15	−15.9264	15.2382	1.8677	0.0328	0.000 00	0.9999	0.0065	−0.0671

$$\gamma_3(S) = \frac{K_S(n,3) - 3K_S(n,1) + 2K_S^3(n,1)}{[1 - K_S^2(n,1)]^{3/2}} ,$$

where

$$K_S(n,r) = \frac{\Gamma[(n-1+r)/2]}{\Gamma[(n-1)/2]} \left(\frac{2}{n-1}\right)^{r/2} .$$

Using similar demonstrations as for the EWMA-S^2 control chart, it can be proven that the constants a_S, b_S and c_S required for the transformation $T_k = a_S + b_S \ln(S_k + c_S)$ can be deduced from the constants $A_S(n)$, $B_S(n)$ and $C_S(n)$ using the following relations

$$b_S = B_S(n) ,$$
$$c_S = C_S(n)\sigma_0 ,$$
$$a_S = A_S(n) - B_S(n)\ln(\sigma_0) .$$

It can also be proven that the initial value Y_0 is equal to

$$Y_0 = A_S(n) + B_S(n)\ln[K_S(n,1) + C_S(n)] .$$

All the constants $A_S(n)$, $B_S(n)$, $C_S(n)$, Y_0, $E(T_k)$ and $\sigma(T_k)$, useful for the EWMA-S control chart, are shown in Table 17.5 for $n \in \{3, \ldots, 15\}$.

Example 17.4: The goal of this example is to show how the EWMA-S control chart behaves in the case of an increase and a decrease in the nominal process variability. We reuse the data in Fig. 17.5 (top and bottom). The corresponding 30 sample standard deviations are plotted in Fig. 17.10, for the two cases (increasing and decreasing). If $n = 5$ and $\sigma_0 = 0.1$, then $b_S = 7.4727$, $c_S = 1.5984 \times 0.1 = 0.159\,84$ and $a_S = -6.8941 - 7.4727 \times \ln(0.1) = 10.3124$. The 30 transformed sample standard deviations are plotted in Fig. 17.11 and the EWMA-S sequence along with the EWMA-S control limits $LCL = 0.000\,14 - 3\sqrt{0.1/1.9} \times 0.9981 = -0.687$ and $UCL = 0.000\,14 + 3\sqrt{0.1/1.9} \times 0.9981 = 0.687$ ($\lambda = 0.1$ and $K = 3$) are plotted in Fig. 17.12. The EWMA-S control chart clearly detects an out-of-control signal at the 25-th subgroup (in the increasing case) and at the 26-th subgroup (in the decreasing case), pointing out that an increase/decrease of the standard deviation occurred.

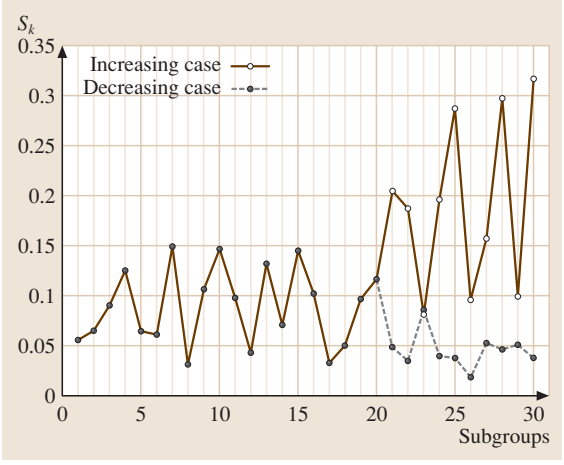


Fig. 17.10 Sample standard deviations S_k corresponding to the data of Fig. 17.5

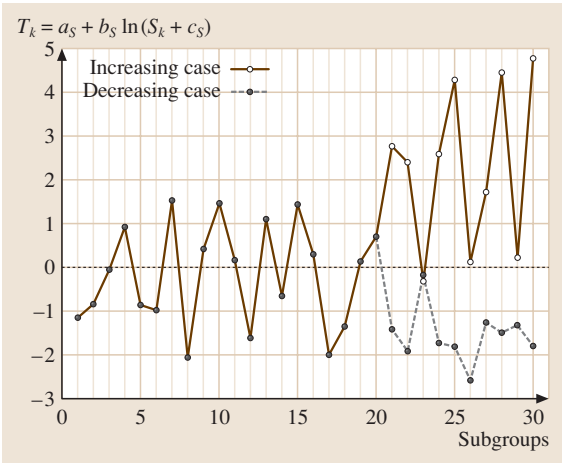


Fig. 17.11 Transformed standard deviations $T_k = a_S + b_S \ln(S_k + c_S)$ corresponding to the data of Fig. 17.5

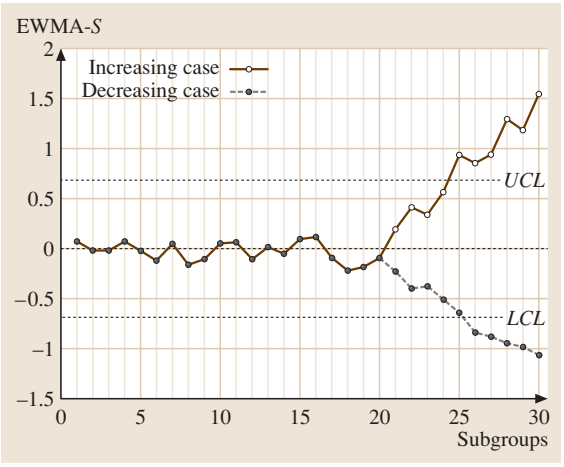


Fig. 17.12 EWMA-S control chart ($\lambda = 0.1$, $K = 3$) corresponding to the data of Fig. 17.5

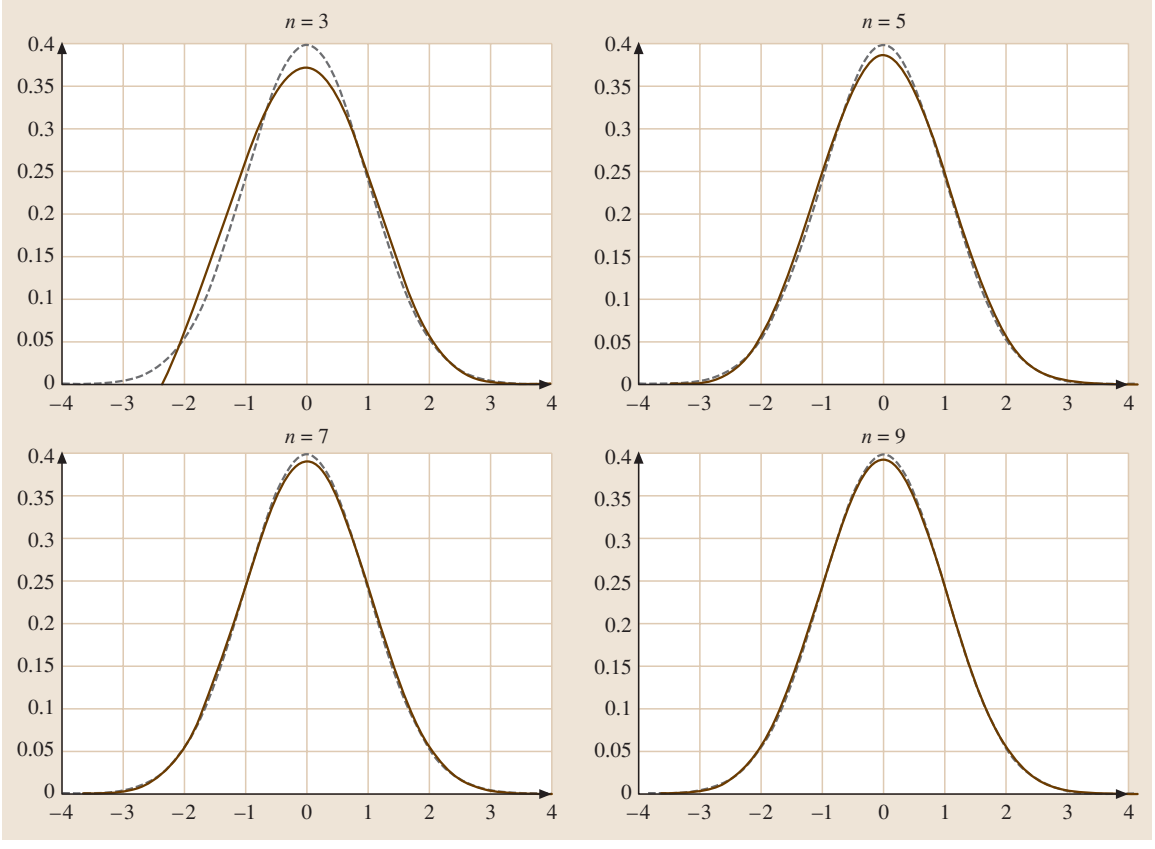


Fig. 17.13 Distribution $f_T(t)$ of $T_k = a_S + b_S \ln(S_k + c_S)$ (*plain line*) compared with the normal (0, 1) distribution (*dotted* line), for $n \in \{3, 5, 7, 9\}$

Table 17.6 Optimal couples (λ^*, K^*) and optimal ARL^* for the EWMA-S control chart, for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{3, 5, 7, 9\}$ and $ARL_0 = 370.4$

EWMA-S																
τ	$n=3$				$n=5$				$n=7$				$n=9$			
	λ^*	K^*	ARL^*	S	λ^*	K^*	ARL^*	S	λ^*	K^*	ARL^*	S	λ^*	K^*	ARL^*	S
0.60	0.10	2.680	14.9	267.0	0.17	2.800	7.9	102.2	0.25	2.867	5.6	45.6	0.29	2.892	4.4	23.6
0.70	0.07	2.590	24.3	363.1	0.12	2.732	13.0	184.5	0.14	2.772	9.1	102.5	0.19	2.834	7.2	62.3
0.80	0.05	2.491	46.2	467.0	0.07	2.594	25.0	308.2	0.08	2.635	17.7	211.8	0.11	2.720	13.9	152.5
0.90	0.05	2.491	132.9	512.1	0.05	2.490	72.2	445.8	0.05	2.490	51.0	384.1	0.05	2.490	40.1	333.1
0.95	0.05	2.491	273.7	463.7	0.05	2.490	183.7	451.0	0.05	2.490	139.2	433.1	0.05	2.490	112.5	414.7
1.05	0.05	2.491	197.7	268.8	0.05	2.490	145.6	253.5	0.05	2.490	115.2	242.1	0.05	2.490	95.8	232.3
1.10	0.05	2.491	94.4	186.4	0.05	2.490	58.8	159.6	0.05	2.490	43.7	140.8	0.05	2.490	35.5	126.2
1.20	0.05	2.491	35.7	90.1	0.05	2.490	22.0	64.5	0.05	2.490	16.7	50.0	0.10	2.696	13.6	40.5
1.30	0.05	2.491	20.4	48.0	0.11	2.713	12.8	30.5	0.17	2.809	9.5	22.0	0.18	2.824	7.7	16.9
1.40	0.05	2.491	14.2	28.5	0.20	2.824	8.6	16.8	0.24	2.862	6.3	11.7	0.32	2.902	5.1	8.8
1.50	0.20	2.793	10.6	18.6	0.32	2.868	6.3	10.5	0.34	2.894	4.6	7.2	0.41	2.918	3.7	5.4
1.60	0.43	2.791	8.1	13.1	0.43	2.870	4.8	7.2	0.49	2.903	3.6	4.9	0.55	2.922	2.9	3.7
1.70	0.56	2.753	6.4	9.8	0.51	2.861	3.9	5.3	0.53	2.901	2.9	3.6	0.63	2.919	2.4	2.8
1.80	0.62	2.734	5.3	7.7	0.60	2.846	3.2	4.2	0.64	2.892	2.4	2.9	0.69	2.916	2.0	2.2
1.90	0.65	2.725	4.5	6.2	0.66	2.835	2.8	3.4	0.70	2.886	2.1	2.4	0.74	2.913	1.7	1.9
2.00	0.67	2.720	3.9	5.2	0.70	2.827	2.4	2.9	0.74	2.882	1.9	2.0	0.74	2.913	1.6	1.6

The distribution $f_T(t)$ of $T_k = a_S + b_S \ln(S_k + c_S)$ (plain line) for $n \in \{3, 5, 7, 9\}$, and the normal (0, 1) distribution (dotted line) are plotted in Fig. 17.13. It is clear that, when n increases, the distribution of T_k becomes closer to the normal (0, 1) distribution, and the lower bound $A_S(n) + B_S(n) \ln[C_S(n)] \to -\infty$. By numerical quadrature, the skewness coefficient $\gamma_3(T_k)$ and the kurtosis coefficient $\gamma_4(T_k)$ of T_k have been computed for $n \in \{3, \ldots, 15\}$; see Table 17.5. As expected, when n increases, T_k becomes more normally distributed, i. e., $\gamma_3(T_k) \to 0$ and $\gamma_4(T_k) \to 0$.

The p.d.f and the c.d.f of the transformed sample standard deviation $T'_k = a_S + b_S \ln(S'_k + c_S)$ after a shift τ are equal to

$$f_{T'}(t) = \frac{1}{\tau B_S(n)} \exp\left(\frac{t - A_S(n)}{B_S(n)}\right) \times f_S\left\{\frac{1}{\tau}\left[\exp\left(\frac{t - A_S(n)}{B_S(n)}\right) - C_S(n)\right] | n\right\},$$

$$F_{T'}(t) = F_S\left\{\frac{1}{\tau}\left[\exp\left(\frac{t - A_S(n)}{B_S(n)}\right) - C_S(n)\right] | n\right\}.$$

The ARL of the EWMA-S control chart can be computed using one of the methods presented in Sect. 17.2.3. The method used for computing the optimal couples (λ^*, K^*) and the corresponding minimal ARL^* for the EWMA-S control chart is exactly the same as the one used for the EWMA-S^2 control chart. The results are shown in Table 17.6. In Table 17.6, we also added, for comparison purpose, the ARL of the classical S control chart. For example, the optimal couple (λ^*, K^*) ensuring the smallest ARL for a shift $\tau = 1.2$ and $n = 7$ is (0.05, 2.490) and the corresponding minimal ARL is $ARL^* = 16.7$, while for the classic S control chart, we have $ARL = 50.0$. Like the EWMA-S^2 control chart, the EWMA-S control chart (with optimized parameters) is more efficient, in terms of ARL, than the S^2 or S control chart. The results of both EWMA-S^2 and EWMA-S control chart are very similar. The main difference is that for the decreasing case ($\tau < 1$) the optimal ARL^*s of the EWMA-S control chart are smaller than those of the EWMA-S^2 control chart, while for the increasing case ($\tau > 1$) the opposite results.

17.3.3 EWMA-*R* Control Chart

Let R_k be the range of the subgroup k, i. e.,

$$R_k = \max(X_{k,1}, \ldots, X_{k,n}) - \min(X_{k,1}, \ldots, X_{k,n}).$$

The EWMA-R control chart proposed by *Castagliola* [17.27] is a natural extension of the EWMA-S^2 control chart where a three-parameter (a_R, b_R, c_R) logarithmic transformation is applied to the range R_k, [i. e. $T_k = a_R + b_R \ln(R_k + c_R)$], instead of the sample vari-

Table 17.7 Expectation $E(R)$, variance $V(R)$ and skewness coefficient $\gamma_3(R)$ of R

n	$E(R)$	$V(R)$	$\gamma_3(R)$
3	1.6926	0.7892	0.6461
4	2.0588	0.7741	0.5230
5	2.3259	0.7466	0.4655
6	2.5344	0.7192	0.4350
7	2.7044	0.6942	0.4176
8	2.8472	0.6721	0.4073
9	2.9700	0.6526	0.4011
10	3.0775	0.6353	0.3976
11	3.1729	0.6199	0.3957
12	3.2585	0.6060	0.3949
13	3.3360	0.5935	0.3949
14	3.4068	0.5822	0.3953
15	3.4718	0.5719	0.3961

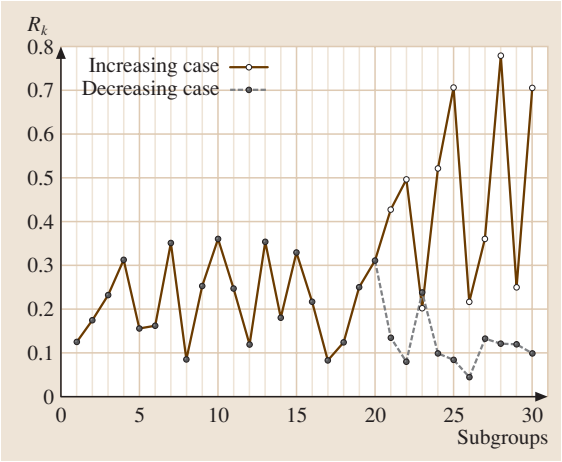


Fig. 17.14 Sample ranges R_k corresponding to the data of Fig. 17.5

ance S_k^2. The control limits of the EWMA-R control chart are given by (17.4) and (17.5) but with different values for $E(T_k)$ and $\sigma(T_k)$. Let R be the range of n independent normal $(\mu_0, 1)$ random variables. The p.d.f. $f_R(r)$ of R is defined for $r \geq 0$ and is given by the well-known relation

$$f_R(r) = n(n-1) \int_{-\infty}^{+\infty} \phi(u)\phi(r+u)[\Phi(r+u) - \Phi(u)]^{n-2} \, \mathrm{d}u \, .$$

In order to compute the constants $A_R(n)$, $B_R(n)$ and $C_R(n)$, the expectation $E(R)$, the variance $V(R)$ and the skewness coefficient $\gamma_3(R)$ of R have been computed by numerical quadrature and are tabulated in Table 17.7. Using similar demonstrations as for the EWMA-S control chart, it can be proven that the constants a_R, b_R and c_R required for the transformation $T_k = a_R + b_R \ln(R_k + c_R)$ can be deduced from the constants $A_R(n)$, $B_R(n)$ and $C_R(n)$ using the following relations

$$b_R = B_R(n) \, ,$$
$$c_R = C_R(n)\sigma_0 \, ,$$

Table 17.8 Constants $A_R(n)$, $B_R(n)$, $C_R(n)$, Y_0, $E(T_k)$, $\sigma(T_k)$, $\gamma_3(T_k)$ and $\gamma_4(T_k)$ for the EWMA-R control chart, for $n \in \{3, \ldots, 15\}$

EWMA-R								
n	$A_R(n)$	$B_R(n)$	$C_R(n)$	Y_0	$E(T_k)$	$\sigma(T_k)$	$\gamma_3(T_k)$	$\gamma_4(T_k)$
3	−6.7191	4.7655	2.4944	0.105	0.000 96	0.9915	0.1370	−0.4429
4	−9.4200	5.8364	3.0385	0.086	0.000 36	0.9961	0.0761	−0.3099
5	−11.1940	6.5336	3.2866	0.077	0.000 19	0.9977	0.0499	−0.2288
6	−12.3056	6.9804	3.3549	0.072	0.000 12	0.9985	0.0361	−0.1767
7	−12.9778	7.2649	3.3202	0.069	0.000 09	0.9989	0.0278	−0.1412
8	−13.3653	7.4446	3.2286	0.067	0.000 07	0.9991	0.0223	−0.1158
9	−13.5689	7.5559	3.1072	0.066	0.000 05	0.9993	0.0185	−0.0969
10	−13.6531	7.6220	2.9715	0.066	0.000 04	0.9994	0.0157	−0.0823
11	−13.6595	7.6576	2.8304	0.065	0.000 04	0.9995	0.0135	−0.0708
12	−13.6151	7.6726	2.6892	0.065	0.000 03	0.9995	0.0118	−0.0615
13	−13.5378	7.6736	2.5508	0.065	0.000 03	0.9996	0.0105	−0.0538
14	−13.4393	7.6647	2.4167	0.065	0.000 03	0.9996	0.0093	−0.0474
15	−13.3276	7.6492	2.2879	0.065	0.000 02	0.9997	0.0084	−0.0420

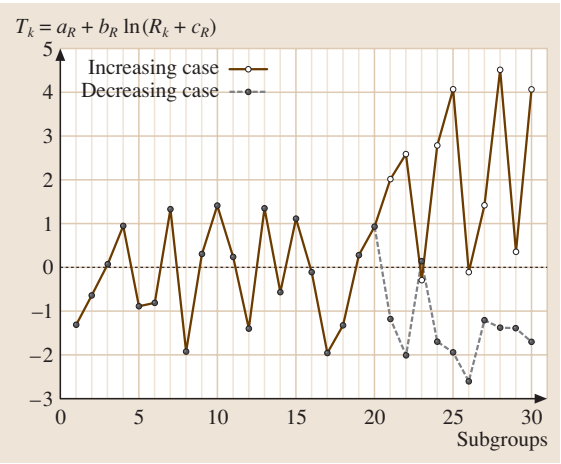


Fig. 17.15 Transformed ranges $T_k = a_R + b_R \ln(R_k + c_R)$ corresponding to the data of Fig. 17.5

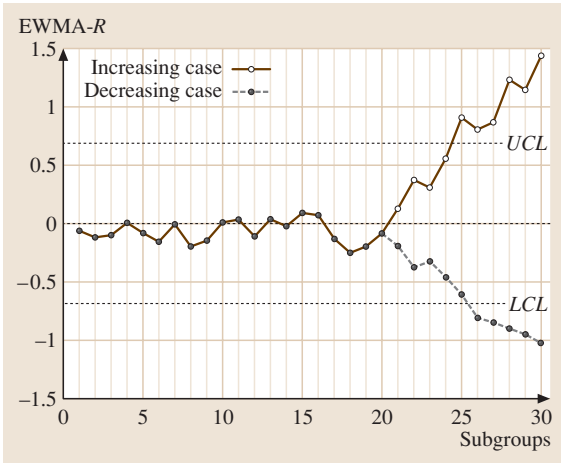


Fig. 17.16 EWMA-R control chart ($\lambda = 0.1$, $K = 3$) corresponding to the data of Fig. 17.5

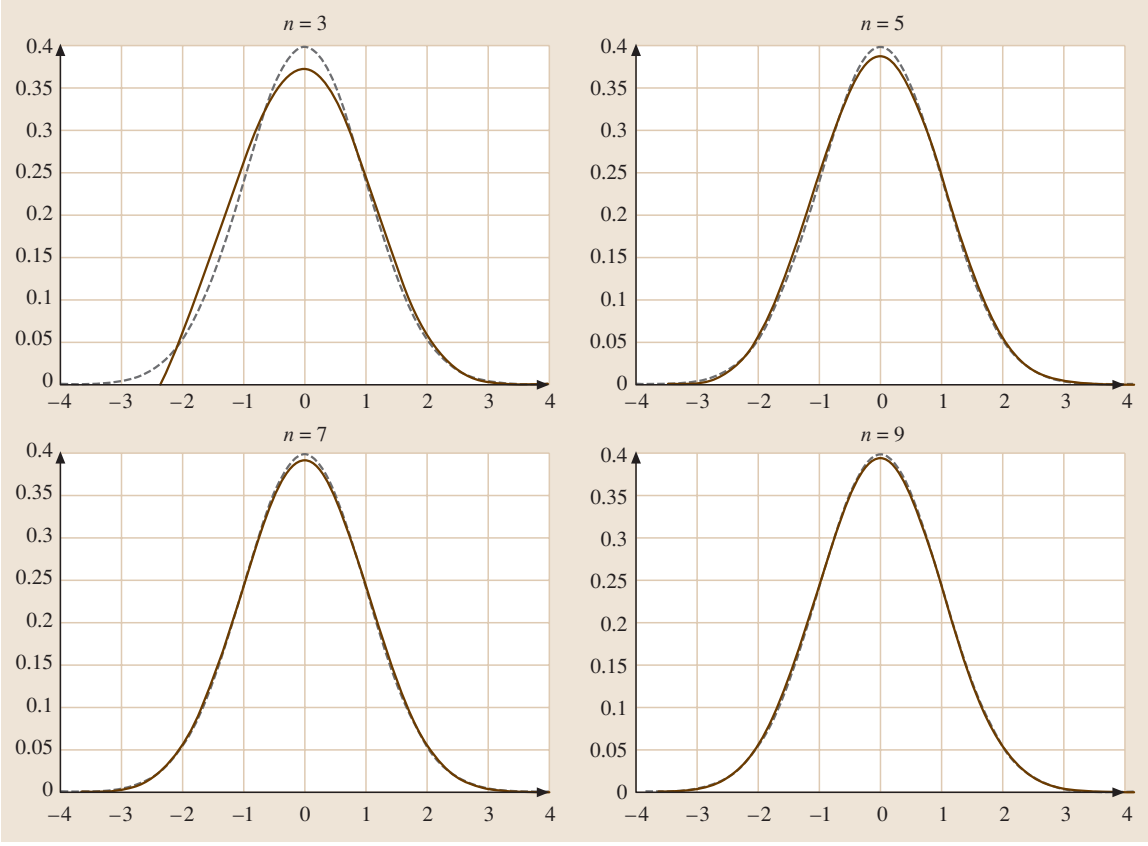


Fig. 17.17 Distribution $f_T(t)$ of $T_k = a_R + b_R \ln(R_k + c_R)$ (*plain line*) compared with the normal (0, 1) distribution (*dotted line*), for $n \in \{3, 5, 7, 9\}$

Table 17.9 Optimal couples (λ^*, K^*) and optimal ARL^* for the EWMA-R control chart, for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{3, 5, 7, 9\}$ and $ARL_0 = 370.4$

EWMA-R																
τ	$n=3$				$n=5$				$n=7$				$n=9$			
	λ^*	K^*	ARL^*	R	λ^*	K^*	ARL^*	R	λ^*	K^*	ARL^*	R	λ^*	K^*	ARL^*	R
0.60	0.10	2.680	15.0	267.0	0.18	2.811	8.2	102.4	0.22	2.854	5.9	46.4	0.27	2.890	4.7	24.8
0.70	0.07	2.591	24.5	363.0	0.11	2.714	13.4	184.7	0.15	2.788	9.7	103.6	0.17	2.817	7.8	64.2
0.80	0.05	2.491	46.6	466.3	0.07	2.595	25.9	307.9	0.08	2.635	19.0	212.7	0.10	2.697	15.4	155.0
0.90	0.05	2.491	133.9	509.6	0.05	2.491	75.0	440.3	0.05	2.491	54.9	378.8	0.05	2.491	44.8	329.6
0.95	0.05	2.491	274.9	461.7	0.05	2.491	188.4	444.5	0.05	2.491	147.5	423.4	0.05	2.491	123.4	403.2
1.05	0.05	2.491	198.1	270.7	0.05	2.491	149.3	261.3	0.05	2.491	122.2	256.5	0.05	2.491	105.4	253.2
1.10	0.05	2.491	94.7	189.0	0.05	2.491	60.5	169.8	0.05	2.491	46.5	158.5	0.05	2.491	39.0	150.4
1.20	0.05	2.491	35.8	92.3	0.05	2.491	22.5	71.7	0.05	2.491	17.6	60.9	0.05	2.491	15.0	53.9
1.30	0.05	2.491	20.4	49.5	0.08	2.633	13.3	34.6	0.12	2.739	10.3	27.6	0.16	2.805	8.7	23.5
1.40	0.05	2.491	14.2	29.5	0.17	2.802	9.0	19.2	0.21	2.847	6.9	14.8	0.25	2.880	5.9	12.3
1.50	0.18	2.782	10.7	19.3	0.26	2.857	6.6	12.0	0.27	2.880	5.1	9.0	0.34	2.913	4.3	7.4
1.60	0.45	2.788	8.2	13.6	0.33	2.873	5.1	8.2	0.37	2.905	4.0	6.1	0.45	2.930	3.4	5.0
1.70	0.56	2.756	6.5	10.1	0.44	2.876	4.1	6.0	0.50	2.912	3.3	4.5	0.48	2.932	2.8	3.6
1.80	0.58	2.750	5.4	7.9	0.55	2.864	3.5	4.7	0.52	2.912	2.7	3.5	0.61	2.934	2.3	2.8
1.90	0.64	2.732	4.6	6.4	0.61	2.854	3.0	3.8	0.63	2.905	2.4	2.8	0.67	2.932	2.0	2.3
2.00	0.67	2.724	4.0	5.3	0.63	2.850	2.6	3.2	0.68	2.901	2.1	2.4	0.71	2.930	1.8	2.0

$$a_R = A_R(n) - B_R(n)\ln(\sigma_0)\,.$$

It can also be proven that the initial value Y_0 is equal to

$$Y_0 = A_R(n) + B_R(n)\ln[K_R(n) + C_R(n)]\,,$$

where $K_R(n)$ is equal to

$$K_R(n) = 2\int_0^{+\infty} 1 - \left[\Phi(x)\right]^n - \left[1-\Phi(x)\right]^n \mathrm{d}x\,.$$

All the constants $A_R(n)$, $B_R(n)$, $C_R(n)$, Y_0, $E(T_k)$ and $\sigma(T_k)$, useful for the EWMA-R control chart, are shown in Table 17.8 for $n \in \{3, \ldots, 15\}$.

Example 17.5: The goal of this example is to show how the EWMA-R control chart behaves in the case of an increase and a decrease in the nominal variability. We reuse the data in Fig. 17.5 (top and bottom). The corresponding 30 sample ranges are plotted in Fig. 17.14, for the two cases (increasing and decreasing). If $n = 5$ and $\sigma_0 = 0.1$, then $b_R = 6.5336$, $c_R = 3.2866 \times 0.1 = 0.328\,66$ and $a_R = -11.1940 - 6.5336 \times \ln(0.1) = 3.8502$. The 30 transformed ranges are plotted in Fig. 17.15 and the EWMA-R sequence along with the EWMA-R control limits $LCL = 0.000\,19 - 3\sqrt{0.1/1.9} \times 0.9977 = -0.686$ and $UCL = 0.000\,19 + 3\sqrt{0.1/1.9} \times 0.9977 = 0.687$ ($\lambda = 0.1$ and $K = 3$) are plotted in Fig. 17.16. The EWMA-R control chart clearly detects an out-of-control signal at the 25-th subgroup (in the increasing case) and at the 26-th subgroup (in the decreasing case), pointing out that an increase/decrease of the dispersion occurred.

The distribution $f_T(t)$ of $T_k = a_R + b_R \ln(R_k + c_R)$ (plain line) for $n \in \{3, 5, 7, 9\}$, and the normal (0, 1) distribution (dotted line) are plotted in Fig. 17.17. It is clear that, when n increases, the distribution of T_k becomes closer to the normal (0, 1) distribution, and the lower bound $A_R(n) + B_R(n)\ln[C_R(n)] \to -\infty$. By numerical quadrature, the skewness coefficient $\gamma_3(T_k)$ and the kurtosis coefficient $\gamma_4(T_k)$ of T_k have been computed for $n \in \{3, \ldots, 15\}$; see Table 17.8. As expected, when n increases, T_k becomes more normally distributed, i. e., $\gamma_3(T_k) \to 0$ and $\gamma_4(T_k) \to 0$.

The p.d.f and the c.d.f of the transformed range $T'_k = a_R + b_R \ln(R'_k + c_R)$ after a shift τ are equal to

$$f_{T'}(t) = \frac{1}{\tau B_R(n)} \exp\left(\frac{t - A_R(n)}{B_R(n)}\right) \times f_R\left\{\frac{1}{\tau}\left[\exp\left(\frac{t - A_R(n)}{B_R(n)}\right) - C_R(n)\right] | n\right\}\,,$$

$$F_{T'}(t) = F_R\left\{\frac{1}{\tau}\left[\exp\left(\frac{t - A_R(n)}{B_R(n)}\right) - C_R(n)\right] | n\right\}\,.$$

The *ARL* of the EWMA-*R* control chart can be computed using one of the methods presented in Sect. 17.2.3. The method used for computing the optimal couples (λ^*, K^*) and the corresponding minimal ARL^* for the EWMA-*R* control chart is exactly the same as the one used for the EWMA-*S* control chart. The results are shown in Table 17.9. In Table 17.9, we also added, for comparison purpose, the *ARL* of the classic *R* control chart. For example, the optimal couple (λ^*, K^*) ensuring the smallest *ARL* for a shift $\tau = 1.2$ and $n = 7$ is (0.05, 2.491) and the corresponding minimal *ARL* is $ARL^* = 17.6$, while for the classic *R* control chart, we have $ARL = 60.9$. The EWMA-*R* control chart (with optimized parameters) is more efficient, in terms of *ARL*, than the *R* control chart, but is slightly less efficient than both the EWMA-S^2 and EWMA-*S* control charts.

17.4 Variable Sampling Interval EWMA Control Charts for Process Dispersion

17.4.1 Introduction

Variable sampling intervals (VSI) control charts are a class of adaptive control charts whose sampling intervals are selected depending on what is observed from the process. VSI control charts have been demonstrated to detect process changes faster than fixed sampling interval (FSI) control charts.

For the VSI charts investigated here, the policy of sampling interval selection is dual: if a point falls into a warning zone near to one of the control limits, the sampling interval to the successive sample should be shorter; otherwise, if the last plotted point is plotted near to the central line, the successive sampling interval can be enlarged, because there is not doubt about a possible out-of-control condition. Most work on developing VSI control charts has been done for the problem of monitoring the mean of the process (see *Reynolds* et al. [17.28], *Reynolds* et al. [17.29], *Runger* and *Pignatiello* [17.30], *Saccucci* et al. [17.31] and *Reynolds* [17.32]). However, fewer works have been done on control charts for the process variance. *Chengular* et al. [17.33] considered a VSI Shewhart chart for monitoring process mean and variance, and very recently, *Reynolds* and *Stoumbos* [17.34] investigated a combination of different control charts for both process mean and variance, using individual observations and variable sampling intervals.

17.4.2 VSI Strategy

Unlike FSI control schemes, the sampling interval between T_k and T_{k+1} depends on the current value of Y_k. A longer sampling interval h_L is used when the control statistic falls within the region $R_L = [LWL, UWL]$, defined as

$$LWL = E(T_k) - W\sqrt{\frac{\lambda}{2-\lambda}}\sigma(T_k)\,,$$

$$UWL = E(T_k) + W\sqrt{\frac{\lambda}{2-\lambda}}\sigma(T_k)\,,$$

where *W* is the width of the *warning limits*, which are always inside the control interval, i.e., $W < K$. Similarly, a short sampling interval h_S is used when the control statistic falls within the region $R_S = [LCL, LWL] \cup [UWL, UCL]$. The process is considered out of control and action should be taken whenever Y_k falls outside the range of the control limits $[LCL, UCL]$. This dual-waiting-time control chart is known to be optimal and easy to implement in practice. *Reynolds* et al. [17.28] have empirically shown that it is optimal to use only two sampling intervals with VSI Shewhart and VSI cumulative sum (CUSUM) control scheme for detecting a specified shift in the process target values. They also gave a general proof for any control scheme that can be represented as a Markov chain. *Saccucci* et al. [17.31] gave a simplified proof based on the theory of Markov chain. In general, these optimality proofs show that the short sampling interval h_S should be made as short as possible, while the long sampling interval h_L should be made as long as possible (i. e. the longest amount of time that is reasonable for the process to run without sampling). As pointed out by *Lucas* and *Saccucci* [17.5], there are practical limitations on how short h_S should be. The value of h_S represents the shortest feasible time interval between subgroups from the process. Any shorter time between subgroups would be impossible due to the amount of time that is required to form the rational subgroups, carry out the inspection, analyze the results from any testing procedures, transport parts and materials, and other delays that would be otherwise inconvenient. So, in this paper we will consider the impact on the expected time until detection, using small but nonzero values of h_S.

Table 17.10 Optimal out-of-control ATS^* of the VSI EWMA-S^2 for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{3, 5\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n = 3$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	17.8	12.7	12.6	12.2	11.9	7.9	7.5	7.7	6.8
0.70	28.5	21.0	20.3	20.0	19.9	14.4	13.4	12.3	11.9
0.80	54.3	41.9	40.3	39.4	39.1	31.7	28.9	27.2	26.7
0.90	155.8	136.1	132.2	129.7	128.9	120.3	113.3	108.8	107.4
0.95	325.7	313.1	310.1	308.0	307.4	303.2	297.8	294.1	293.0
1.05	153.6	146.1	144.8	144.0	143.8	140.1	137.9	136.5	136.1
1.10	64.9	55.4	54.0	53.0	52.8	47.8	45.2	43.6	43.1
1.20	21.6	14.7	14.0	13.6	13.4	9.2	7.9	7.1	6.9
1.30	11.6	6.8	6.5	6.4	6.3	3.0	2.5	2.2	2.1
1.40	7.8	4.3	4.1	4.0	4.0	1.5	1.2	1.1	1.0
1.50	5.9	3.1	3.0	3.0	3.0	0.9	0.7	0.7	0.7
1.60	4.8	2.5	2.4	2.4	2.4	0.6	0.5	0.5	0.5
1.70	4.0	2.1	2.0	2.0	2.0	0.5	0.4	0.4	0.4
1.80	3.5	1.8	1.8	1.8	1.8	0.4	0.4	0.4	0.4
1.90	3.1	1.6	1.6	1.6	1.6	0.3	0.3	0.3	0.3
2.00	2.9	1.4	1.4	1.4	1.4	0.3	0.3	0.3	0.3
	$n = 5$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	9.0	6.8	5.9	5.7	5.7	4.5	3.2	2.4	2.4
0.70	14.6	11.0	10.2	10.2	10.2	7.7	5.9	5.6	5.5
0.80	28.2	21.3	20.8	20.3	20.2	15.6	14.5	13.8	13.6
0.90	81.0	67.6	65.3	63.9	63.5	56.9	52.8	50.2	49.4
0.95	202.9	189.9	187.0	185.1	184.5	179.4	174.3	170.8	169.8
1.05	121.4	111.1	109.1	107.9	107.5	102.9	99.4	97.2	96.5
1.10	44.8	34.8	33.3	32.4	32.1	26.9	24.1	22.4	21.9
1.20	15.3	9.3	8.8	8.5	8.4	4.6	3.6	3.1	3.0
1.30	8.8	4.8	4.6	4.5	4.5	1.7	1.3	1.1	1.1
1.40	6.2	3.3	3.2	3.1	3.1	1.0	0.8	0.7	0.7
1.50	4.8	2.5	2.4	2.4	2.4	0.7	0.5	0.5	0.5
1.60	4.0	2.0	2.0	2.0	2.0	0.5	0.4	0.4	0.4
1.70	3.5	1.8	1.7	1.7	1.7	0.4	0.4	0.3	0.3
1.80	2.9	1.6	1.5	1.5	1.5	0.3	0.3	0.3	0.3
1.90	2.5	1.4	1.4	1.4	1.4	0.3	0.3	0.3	0.3
2.00	2.3	1.3	1.3	1.3	1.3	0.3	0.3	0.3	0.3

Table 17.11 Optimal out-of-control ATS^* of the VSI EWMA-S^2 for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{7, 9\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n = 7$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	6.2	4.6	4.6	3.5	3.5	3.0	3.0	1.2	1.1
0.70	10.0	7.5	7.0	6.5	6.5	5.2	4.6	3.1	3.0
0.80	19.5	15.1	14.1	14.0	14.0	11.1	9.2	8.8	8.7
0.90	56.8	46.5	45.0	44.0	43.7	38.4	35.5	33.7	33.2
0.95	150.6	137.7	135.1	133.2	132.7	127.5	122.7	119.4	118.4
1.05	99.3	95.2	92.9	91.3	90.9	79.3	75.4	72.8	72.1
1.10	34.9	27.3	25.7	24.8	24.5	18.3	15.7	14.2	13.7
1.20	12.4	7.8	7.3	7.0	7.0	3.4	2.5	2.1	2.0
1.30	7.4	4.3	4.0	4.0	3.9	1.4	1.0	0.9	0.9
1.40	5.3	3.0	2.9	2.8	2.8	0.8	0.6	0.6	0.6
1.50	4.2	2.3	2.2	2.2	2.2	0.6	0.5	0.4	0.4
1.60	3.2	1.9	1.9	1.9	1.9	0.5	0.4	0.4	0.4
1.70	2.6	1.6	1.6	1.6	1.6	0.4	0.3	0.3	0.3
1.80	2.2	1.5	1.5	1.4	1.4	0.3	0.3	0.3	0.3
1.90	2.0	1.3	1.3	1.3	1.3	0.3	0.3	0.3	0.3
2.00	1.8	1.2	1.2	1.2	1.2	0.3	0.2	0.2	0.2
	$n = 9$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	4.7	3.5	3.7	2.5	2.5	2.4	2.8	0.7	0.7
0.70	7.7	5.8	5.9	4.7	4.7	3.9	4.1	2.0	1.9
0.80	15.1	11.5	10.8	10.5	10.5	8.4	7.4	6.1	6.1
0.90	44.7	36.3	34.9	34.1	33.9	29.6	27.0	25.7	25.3
0.95	120.9	108.6	106.1	104.4	103.9	98.7	94.3	91.3	90.3
1.05	84.0	72.7	70.5	69.0	68.5	63.6	59.6	56.9	56.1
1.10	29.0	20.6	19.3	18.5	18.3	13.8	11.5	10.1	9.7
1.20	10.7	6.3	5.9	5.7	5.6	2.8	2.0	1.6	1.5
1.30	6.6	3.6	3.4	3.3	3.3	1.2	0.9	0.8	0.7
1.40	4.7	2.6	2.5	2.4	2.4	0.8	0.6	0.5	0.5
1.50	3.4	2.0	2.0	1.9	1.9	0.5	0.4	0.4	0.4
1.60	2.7	1.7	1.6	1.6	1.6	0.4	0.3	0.3	0.3
1.70	2.2	1.5	1.4	1.4	1.4	0.3	0.3	0.3	0.3
1.80	1.9	1.3	1.3	1.3	1.3	0.3	0.3	0.3	0.3
1.90	1.6	1.2	1.2	1.2	1.2	0.3	0.2	0.2	0.2
2.00	1.5	1.1	1.1	1.1	1.1	0.2	0.2	0.2	0.2

17.4.3 Average Time to Signal for a VSI Control Chart

If the *ARL* is a useful tool for comparing the performance of various control charts, this indicator cannot be used in the case of VSI-type control charts since the interval between two consecutive samples is not constant. As a consequence, it is common to use the *average time to signal* (*ATS*): when the process is in-control and remains in this state, it is desirable to have

Table 17.12 Optimal h_L^* values of the VSI EWMA-S^2 for $n \in \{3, 5, 7, 9\}$, $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n=3$								$n=5$							
	$h_S=0.5$				$h_S=0.1$				$h_S=0.5$				$h_S=0.1$			
	W				W				W				W			
τ	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15
0.60	1.30	1.59	2.67	4.46	1.52	2.16	3.94	7.23	1.28	1.64	2.61	4.39	1.51	2.15	3.92	7.13
0.70	1.28	1.60	2.69	4.56	1.54	2.06	4.00	7.23	1.29	1.57	2.62	4.40	1.51	2.15	3.91	7.10
0.80	1.27	1.56	2.46	4.65	1.48	2.01	3.63	7.56	1.27	1.60	2.68	4.51	1.49	2.05	4.00	7.27
0.90	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
0.95	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.05	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.10	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.20	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.30	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.40	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.50	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.60	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.70	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.80	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
1.90	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
2.00	1.27	1.57	2.48	4.70	1.49	2.03	3.67	7.66	1.27	1.57	2.49	4.71	1.49	2.03	3.68	7.67
	$n=7$								$n=9$							
	$h_S=0.5$				$h_S=0.1$				$h_S=0.5$				$h_S=0.1$			
	W				W				W				W			
τ	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15
0.60	1.28	1.57	2.60	4.35	1.51	2.02	3.92	7.11	1.27	1.62	2.58	4.32	1.50	2.03	3.87	7.03
0.70	1.28	1.57	2.60	4.35	1.50	2.02	3.87	7.06	1.27	1.62	2.58	4.32	1.49	2.03	3.85	6.98
0.80	1.26	1.58	2.62	4.41	1.51	2.02	3.88	7.05	1.28	1.57	2.59	4.35	1.50	2.03	3.86	7.00
0.90	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.55	2.44	4.61	1.48	2.09	4.06	7.50
0.95	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.05	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.10	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.20	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.30	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.40	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.50	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.60	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.70	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.80	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
1.90	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63
2.00	1.27	1.57	2.48	4.69	1.49	2.03	3.67	7.65	1.27	1.57	2.48	4.68	1.48	2.02	3.66	7.63

Table 17.13 Optimal couples (λ^*, K^*) of the VSI EWMA-S^2 for $n \in \{3, 5\}$, $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n = 3$															
	$h_S = 0.5$								$h_S = 0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.13	2.742	0.12	2.726	0.11	2.709	0.13	2.742	0.16	2.779	0.19	2.808	0.16	2.779	0.13	2.742
0.70	0.09	2.669	0.09	2.669	0.09	2.669	0.08	2.643	0.12	2.726	0.11	2.709	0.11	2.709	0.13	2.742
0.80	0.05	2.548	0.06	2.583	0.06	2.583	0.06	2.583	0.06	2.583	0.06	2.583	0.06	2.583	0.06	2.583
0.90	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
0.95	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.05	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.10	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.20	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.30	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.40	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.50	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.60	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.70	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.80	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
1.90	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
2.00	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548	0.05	2.548
	$n = 5$															
	$h_S = 0.5$								$h_S = 0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.21	2.808	0.22	2.812	0.24	2.820	0.24	2.820	0.31	2.833	0.23	2.816	0.31	2.833	0.31	2.833
0.70	0.15	2.763	0.15	2.763	0.17	2.782	0.17	2.782	0.19	2.797	0.21	2.808	0.24	2.820	0.24	2.820
0.80	0.09	2.663	0.09	2.663	0.09	2.663	0.09	2.663	0.10	2.687	0.11	2.707	0.10	2.687	0.10	2.687
0.90	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
0.95	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.05	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.10	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.20	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.30	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.40	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.50	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.60	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.70	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.80	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
1.90	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515
2.00	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515	0.05	2.515

a large *ATS* since it represents the expected value of the elapsed time between two consecutive false alarms. Otherwise, if the characteristic of the process has shifted, it is desirable to have an *ATS* that is as small as possible. Since it represents the expected value of the elapsed time between the occurence of a special cause,

Table 17.14 Optimal couples (λ^*, K^*) of the VSI EWMA-S^2 for $n \in \{7, 9\}$, $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n=7$															
	$h_S = 0.5$								$h_S = 0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.33	2.849	0.15	2.771	0.28	2.843	0.28	2.843	0.41	2.848	0.15	2.771	0.41	2.848	0.41	2.848
0.70	0.20	2.813	0.15	2.771	0.20	2.813	0.24	2.833	0.28	2.843	0.15	2.771	0.28	2.843	0.33	2.849
0.80	0.11	2.710	0.11	2.710	0.13	2.745	0.13	2.745	0.14	2.759	0.14	2.759	0.18	2.800	0.18	2.800
0.90	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
0.95	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.05	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.10	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.20	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.30	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.40	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.50	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.60	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.70	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.80	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
1.90	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
2.00	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506	0.05	2.506
	$n=9$															
	$h_S = 0.5$								$h_S = 0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.33	2.865	0.34	2.866	0.33	2.865	0.33	2.865	0.46	2.864	0.12	2.733	0.41	2.868	0.54	2.854
0.70	0.27	2.854	0.27	2.854	0.27	2.854	0.27	2.854	0.33	2.865	0.12	2.733	0.38	2.868	0.42	2.867
0.80	0.15	2.777	0.12	2.733	0.17	2.798	0.17	2.798	0.17	2.798	0.12	2.733	0.23	2.839	0.23	2.839
0.90	0.05	2.502	0.06	2.555	0.06	2.555	0.06	2.555	0.05	2.502	0.07	2.599	0.07	2.599	0.06	2.555
0.95	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.05	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.10	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.20	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.30	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.40	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.50	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.60	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.70	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.80	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
1.90	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502
2.00	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502	0.05	2.502

i. e. the transition of the process to an out-of-control state, and the signal from the control chart. For an FSI model, the *ATS* is a multiple of the *ARL* since the time h_F between samples is fixed. Thus, in this case we have

$$ATS^{FSI} = h_F \times ARL^{FSI} .$$

For a VSI model, the *ATS* depends on both the number of samples to signal (the *ARL*) and the sampling frequency, which is variable,

$$ATS^{VSI} = E(h) \times ARL^{VSI} ,$$

where $E(h)$ represents the expected value of the sampling interval. For given λ and K, the value of $E(h)$ depends on W, h_S and h_L. For fixed values of h_S and $E(h)$, with $h_S < E(h)$, there is (see *Reynolds* [17.35]) a one-to-one correspondence between W and h_L such that, if the first one decreases, the second one has to increase, and conversely. If we have to compare the out-of-control *ATS* performance of two FSI-type control charts, we only need to define them with the same in-control $ARL_Y = ARL_0$. For VSI-type control charts, this is a little more complex because, if one control chart samples the process more frequently than another, it will necessarily detect a shift in the process sooner than the other one. Consequently, if we have to compare the out-of-control *ATS* performance of two VSI-type control charts, we need to define them with the same in-control $ARL_Y = ARL_0$ *and* the same in-control average sampling interval $E_0(h)$. Because, for *FSI*-type control charts, we have $h_S = h_L = h_F = 1$ time units, the in-control average sampling interval is chosen to be $E_0(h) = 1$. This ensures that we have the same $ATS_Y = ATS_0$ for both FSI- and VSI-type control charts.

17.4.4 Performance of the VSI EWMA-S^2 Control Chart

The performance of the VSI EWMA-S^2 control chart has been investigated by *Castagliola* et al. [17.36]. The *ATS* of the VSI EWMA-S^2 control chart can be computed using the second approach presented in Sect. 17.2.3, where the element g_j of the $(p, 1)$ vector $\boldsymbol{g}$ is defined by

$$g_j = \begin{cases} h_L & \text{if } LWL < H_j < UWL \\ h_S & \text{otherwise} \end{cases} .$$

The optimization scheme of the VSI EWMA-S^2 control chart consists of finding the optimal combination (λ^*, K^*, h_L^*) that give the same in-control $ATS_Y = ATS_0$ (i. e. the *ATS* when the process is functioning at the nominal variability $\sigma = \sigma_0$ or equivalently $\tau = 1$) and then, for predefined values of τ, W and h_S, find the unique combination (λ^*, K^*, h_L^*) which yields the smallest possible out-of-control $ATS_Y = ATS^*$, subject to the constraint $E_0(h) = 1$. The minimal *ATS* values achieved using the optimal VSI model are summarized in Table 17.10 and Table 17.11 for shift $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$ and $n \in \{3, 5, 7, 9\}$. The values used for h_S are 0.5 and 0.1 time units. For comparison purposes, Tables 17.10 and 17.11 also show the minimal *ATS* of the FSI EWMA-S^2 (column $h_L = 1$). As expected, the results clearly indicate that the VSI model outperforms the FSI scheme for all considered shifts of variability since the VSI model gives a signal earlier than the FSI model. For example, for $n = 5$, $h_S = 0.5$ and $W = 0.6$ we have $ATS^* = 65.3$ when $\tau = 0.9$, while for the FSI model we have $ATS^* = 81.0$. When $\tau = 1.2$, we have $ATS^* = 8.8$ for the VSI EWMA-S^2 control chart, while for the FSI model we have $ATS^* = 15.3$. We can also notice that the performance in term of *ATS* is improved when a smaller short sampling interval h_S is considered and, for a selected value of h_S, the *ATS* is improved as the value of W decreases. Table 17.12 shows the optimal long sampling interval h_L^* for several combinations of n, W, h_S and τ. For a defined value of h_S, when W decreases it is no surprise to remark that the length of the long sampling interval h_L^* increases. Finally, Tables 17.13 and 17.14 summarize the optimal couples (λ^*, K^*).

Example 17.6: The goal of this example is to illustrate the use of the VSI EWMA-S^2 control chart using a simulated process. The sample size is assumed to be $n = 5$. We assume that during the first 20 units of time the data are generated according to a normal $(20, 0.1)$ distribution (corresponding to an in-control process) while, after the first 20 units of time, the data are generated according to a normal $(20, 0.13)$ distribution (the nominal process standard deviation σ_0 has increased by a factor of 1.3). If $n = 5$ and $\sigma_0 = 0.1$, then we deduce from Table 17.3: $b_{S^2} = 2.3647$, $c_{S^2} = 0.5979 \times 0.1^2 = 0.005\,979$ and $a_{S^2} = -0.8969 - 2 \times 2.3647 \times \ln(0.1) = 9.9929$. The FSI EWMA chart has been designed by considering $\lambda = 0.05$, $K = 2.515$ and $h_F = 1$; the VSI EWMA was implemented by considering the same couple (λ, K) and $h_S = 0.5$, $h_L = 1.27$, $W = 0.9$. This choice ensures that both the FSI and VSI EWMA-S^2 control charts are designed to have the

Table 17.15 Subgroup number, sampling interval (h_S or h_L), total elapsed time from the start of the simulation and statistics S_k^2, T_k and Y_k

Subgroup	Sampling interval	Total time	S_k^2	T_k	Y_k
1	0.50	0.50	0.003 38	− 1.052	0.148
2	0.50	1.00	0.003 15	− 1.112	0.085
3	1.27	2.27	0.018 55	1.225	0.142
4	1.27	3.54	0.002 54	− 1.275	0.071
5	1.27	4.81	0.004 52	− 0.782	0.028
6	1.27	6.08	0.003 76	− 0.959	− 0.021
7	1.27	7.35	0.009 34	0.112	− 0.014
8	1.27	8.62	0.008 53	− 0.017	− 0.014
9	1.27	9.89	0.017 23	1.094	0.041
10	1.27	11.16	0.008 61	− 0.003	0.039
11	1.27	12.43	0.027 72	1.976	0.136
12	1.27	13.70	0.014 22	0.765	0.167
13	0.50	14.20	0.016 96	1.067	0.212
14	0.50	14.70	0.010 37	0.266	0.215
15	0.50	15.20	0.009 56	0.146	0.211
16	0.50	15.70	0.005 31	− 0.610	0.170
17	0.50	16.20	0.008 97	0.053	0.164
18	0.50	16.70	0.008 21	− 0.070	0.153
19	0.50	17.20	0.009 76	0.176	0.154
20	0.50	17.70	0.016 04	0.970	0.195
21	0.50	18.20	0.005 95	− 0.479	0.161
22	0.50	18.70	0.010 25	0.249	0.165
23	0.50	19.20	0.007 75	− 0.148	0.150
24	0.50	19.70	0.011 63	0.441	0.164
25	0.50	20.20	0.012 12	0.507	0.181
26	0.50	20.70	0.030 40	2.157	0.280
27	0.50	21.20	0.019 89	1.351	0.334
28	0.50	21.70	0.016 07	0.973	0.366
29	0.50	22.20	0.013 19	0.641	0.379
30	0.50	22.70	0.001 53	− 1.573	0.282
31	0.50	23.20	0.009 95	0.203	0.278
32	0.50	23.70	0.020 19	1.378	0.333
33	0.50	24.20	0.008 90	0.043	0.318
34	0.50	24.70	0.025 98	1.850	0.395
35	0.50	25.20	0.019 70	1.333	0.442

same in control $ARL_0 = 370.4$ and are designed to optimally detect a $\tau = 1.5$ shift for the process variability. The control and warning limits are $LCL = 0.0075 - 2.515\sqrt{0.05/1.95} \times 0.967 = -0.382$, $UCL = 0.0075 + 2.515\sqrt{0.05/1.95} \times 0.967 = 0.397$, $LWL = 0.0075 - 0.9\sqrt{0.05/1.95} \times 0.967 = -0.132$ and $UWL = 0.0075 + 0.9\sqrt{0.05/1.95} \times 0.967 = 0.147$. In Table 17.15 we summarize the results of this simulation, i. e. the subgroup number, the sampling interval (h_S or h_L) used for each sample, the total elapsed time from the start of the process simulation and the statistics S_k^2, T_k and Y_k. In Fig. 17.18 (top), we plot the VSI EWMA-S^2 control chart (i. e. the Y_ks) corresponding to our data. As we can see, the VSI EWMA-S^2 control chart clearly detects an out-of-control signal after 25.2 units of time (35-th subgroup), pointing out that an increase of the variance occurred. In Fig. 17.18 (bottom), we plotted the FSI EWMA-S^2 control chart using the same data but

Table 17.16 Optimal out-of-control ATS^* of the VSI EWMA-R for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{3, 5\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n = 3$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	15.0	11.1	11.3	10.2	10.1	7.7	7.7	5.5	5.2
0.70	24.5	18.9	18.7	17.8	17.8	13.9	13.7	11.3	11.2
0.80	46.6	38.0	36.4	35.8	35.6	31.0	27.9	26.9	26.7
0.90	133.9	121.6	118.6	116.9	116.4	112.0	106.6	103.5	102.6
0.95	274.9	267.7	265.9	264.8	264.5	262.4	259.2	257.2	256.6
1.05	198.1	192.0	190.1	189.0	188.7	186.9	183.5	181.5	180.9
1.10	94.7	86.4	83.9	82.5	82.1	79.4	74.8	72.3	71.6
1.20	35.8	29.2	27.2	26.2	25.9	23.7	20.0	18.3	17.8
1.30	20.4	15.7	14.1	13.4	13.3	11.8	8.8	7.7	7.4
1.40	14.2	10.6	9.2	8.8	8.6	7.6	5.1	4.3	4.1
1.50	10.7	8.1	6.8	6.5	6.4	5.7	3.5	2.8	2.7
1.60	8.2	6.6	5.4	5.2	5.1	4.6	2.6	2.1	1.9
1.70	6.5	5.6	4.5	4.3	4.2	3.9	2.0	1.6	1.5
1.80	5.4	4.7	3.9	3.7	3.7	3.5	1.6	1.3	1.2
1.90	4.6	4.1	3.4	3.2	3.1	3.1	1.4	1.1	1.0
2.00	4.0	3.6	3.1	2.8	2.7	2.9	1.2	1.0	0.9
	$n = 5$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	8.2	6.0	6.1	5.6	4.9	4.0	4.2	3.4	1.9
0.70	13.4	10.0	10.1	9.2	8.9	7.0	7.0	5.7	4.7
0.80	25.9	20.2	20.1	19.0	19.1	15.4	15.1	12.8	12.8
0.90	75.0	64.7	63.4	61.5	61.2	56.4	54.1	50.7	50.2
0.95	188.4	179.1	177.4	175.5	175.1	171.7	168.6	165.2	164.5
1.05	149.3	140.9	139.1	137.0	136.6	134.0	130.9	127.2	126.4
1.10	60.5	52	50.5	48.3	47.9	45.1	42.4	38.4	37.7
1.20	22.5	17.3	16.6	14.9	14.7	13.1	11.9	8.8	8.5
1.30	13.3	9.9	9.6	8.1	8.0	7.2	6.6	3.9	3.7
1.40	9.0	6.9	6.9	5.5	5.4	5.0	4.7	2.3	2.1
1.50	6.6	5.2	5.4	4.2	4.1	4.0	3.8	1.6	1.5
1.60	5.1	4.2	4.4	3.3	3.2	3.3	3.3	1.2	1.1
1.70	4.1	3.5	3.7	2.8	2.6	2.9	3.0	0.9	0.9
1.80	3.5	3.0	3.2	2.4	2.1	2.5	2.8	0.8	0.7
1.90	3.0	2.6	2.9	2.1	1.8	2.3	2.6	0.6	0.6
2.00	2.6	2.4	2.7	1.9	1.5	2.2	2.5	0.6	0.5

Table 17.17 Optimal out-of-control ATS^* of the VSI EWMA-R for $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $n \in \{7, 9\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n = 7$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	5.9	4.3	4.5	4.4	3.3	2.9	3.2	2.6	1.1
0.70	9.7	7.2	7.3	7.0	6.1	4.9	5.1	4.6	2.8
0.80	19.0	14.5	14.5	13.5	13.3	10.7	10.7	9.1	8.2
0.90	54.9	46.2	45.4	43.9	43.7	39.1	37.7	35.0	34.8
0.95	147.5	137.4	135.6	133.7	133.3	129.3	126.2	122.6	121.9
1.05	122.2	113.0	111.1	108.9	108.4	105.5	102.2	98.2	97.4
1.10	46.5	38.5	37.2	35.2	34.8	32.1	29.8	26.0	25.4
1.20	17.6	13.2	12.7	11.2	11.0	9.8	8.9	6.0	5.7
1.30	10.3	7.8	7.7	6.3	6.2	5.7	5.3	2.7	2.5
1.40	6.9	5.4	5.5	4.3	4.3	4.1	3.9	1.6	1.5
1.50	5.1	4.1	4.3	3.3	3.1	3.2	3.3	1.1	1.0
1.60	4.0	3.3	3.5	2.6	2.3	2.7	2.9	0.8	0.8
1.70	3.3	2.8	3.0	2.2	1.9	2.3	2.7	0.7	0.6
1.80	2.7	2.4	2.7	1.9	1.6	2.1	2.5	0.5	0.5
1.90	2.4	2.2	2.4	1.7	1.4	2.0	2.4	0.5	0.4
2.00	2.1	2.0	2.3	1.5	1.2	1.9	2.3	0.4	0.4
	$n = 9$								
	FSI	VSI							
	$h_L = 1$	$h_S = 0.5$				$h_S = 0.1$			
		W				W			
τ		0.9	0.6	0.3	0.1	0.9	0.6	0.3	0.1
0.60	4.7	3.5	3.7	3.8	2.5	2.4	2.8	2.1	0.8
0.70	7.8	5.8	5.9	6.0	4.6	4.0	4.2	3.9	2.0
0.80	15.4	11.7	11.7	11.1	10.5	8.5	8.5	7.6	6.1
0.90	44.8	37.0	36.5	35.2	35.2	30.7	29.8	27.4	27.3
0.95	123.4	112.8	111.2	109.3	109.0	104.6	101.6	98.2	97.6
1.05	105.4	96.2	94.3	92.1	91.7	88.5	85.2	81.1	80.4
1.10	39.0	31.8	30.7	28.7	28.4	25.8	23.8	20.3	19.7
1.20	15.0	11.3	11.0	9.4	9.3	8.3	7.6	4.9	4.6
1.30	8.7	6.6	6.6	5.3	5.3	4.9	4.7	2.3	2.1
1.40	5.9	4.5	4.7	3.7	3.5	3.4	3.6	1.4	1.2
1.50	4.3	3.4	3.7	2.8	2.5	2.7	3.1	0.9	0.8
1.60	3.4	2.8	3.1	2.3	1.9	2.3	2.7	0.7	0.6
1.70	2.8	2.4	2.7	1.9	1.6	2.1	2.5	0.6	0.5
1.80	2.3	2.1	2.4	1.7	1.3	1.9	2.4	0.5	0.4
1.90	2.0	1.9	2.2	1.5	1.2	1.8	2.3	0.4	0.3
2.00	1.8	1.8	2.1	1.4	1.0	1.7	2.2	0.3	0.3

assuming a fixed sampling rate $h_F = 1$. The difference between the FSI and the VSI EWMA-S^2 control chart appears clearly and, in this example, the difference in terms of detection time is 9.8 units of time.

Table 17.18 Optimal h_L^* values of the VSI EWMA-R for $n \in \{3, 5, 7, 9\}$, $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n=3$								$n=5$							
	$h_S=0.5$				$h_S=0.1$				$h_S=0.5$				$h_S=0.1$			
	W				W				W				W			
τ	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15
0.60	1.28	1.55	2.59	5.05	1.50	2.14	3.86	8.28	1.27	1.61	2.59	4.96	1.54	2.09	3.86	8.09
0.70	1.26	1.60	2.64	5.15	1.51	2.00	3.89	8.34	1.28	1.54	2.59	5.01	1.48	2.11	3.86	8.12
0.80	1.25	1.53	2.39	3.97	1.49	2.07	4.05	8.70	1.26	1.56	2.63	5.14	1.51	1.99	3.89	8.35
0.90	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.24	1.52	2.38	3.95	1.44	1.93	3.48	6.30
0.95	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.24	1.52	2.38	3.95	1.44	1.93	3.48	6.30
1.05	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.24	1.52	2.38	3.95	1.44	1.93	3.48	6.30
1.10	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.24	1.52	2.38	3.95	1.44	1.93	3.48	6.30
1.20	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.24	1.52	2.38	3.95	1.44	1.93	3.48	6.30
1.30	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.24	1.52	2.38	3.95	1.44	1.93	3.48	6.30
1.40	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.28	1.52	2.38	3.95	1.44	1.93	3.48	6.30
1.50	1.25	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.27	1.62	2.38	3.95	1.44	1.93	3.48	6.30
1.60	1.27	1.53	2.39	3.97	1.44	1.95	3.51	6.35	1.30	1.61	2.59	4.94	1.48	1.93	3.48	6.30
1.70	1.28	1.60	2.39	3.97	1.44	1.95	3.51	6.35	1.30	1.61	2.59	4.95	1.48	1.93	3.48	6.30
1.80	1.29	1.60	2.39	3.97	1.44	1.95	3.51	6.35	1.31	1.62	2.59	4.97	1.48	1.93	3.48	6.30
1.90	1.29	1.60	2.59	5.22	1.44	1.95	3.51	6.35	1.31	1.62	2.59	4.97	1.54	1.93	3.48	6.30
2.00	1.29	1.60	2.59	5.36	1.44	1.95	3.51	6.35	1.28	1.63	2.59	4.97	1.55	1.93	3.48	6.30
	$n=7$								$n=9$							
	$h_S=0.5$				$h_S=0.1$				$h_S=0.5$				$h_S=0.1$			
	W				W				W				W			
τ	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15	0.9	0.6	0.3	0.15
0.60	1.30	1.60	2.60	4.90	1.53	3.88	3.88	7.98	1.29	1.59	2.61	4.86	1.52	2.06	3.90	7.93
0.70	1.27	1.61	2.60	4.91	1.54	3.88	3.88	8.02	1.30	1.60	2.61	4.90	1.53	2.07	3.90	7.95
0.80	1.28	1.54	2.60	5.04	1.50	3.88	3.88	8.15	1.28	1.62	2.61	4.98	1.49	2.10	3.90	8.11
0.90	1.24	1.52	2.37	3.94	1.44	3.47	3.47	6.29	1.24	1.52	2.37	3.93	1.48	2.04	4.00	8.59
0.95	1.24	1.52	2.37	3.94	1.44	3.47	3.47	6.29	1.24	1.52	2.37	3.93	1.43	1.93	3.47	6.27
1.05	1.24	1.52	2.37	3.94	1.44	3.47	3.47	6.29	1.24	1.52	2.37	3.93	1.43	1.93	3.47	6.27
1.10	1.24	1.52	2.37	3.94	1.44	3.47	3.47	6.29	1.24	1.52	2.37	3.93	1.43	1.93	3.47	6.27
1.20	1.24	1.52	2.37	3.94	1.44	3.47	3.47	6.29	1.26	1.52	2.37	3.93	1.43	1.93	3.47	6.27
1.30	1.28	1.52	2.37	3.94	1.44	3.47	3.47	6.29	1.27	1.54	2.65	5.17	1.49	1.93	3.47	6.27
1.40	1.30	1.60	2.60	4.99	1.49	3.47	3.47	6.29	1.30	1.60	2.61	4.91	1.47	1.93	3.47	6.27
1.50	1.29	1.60	2.60	4.92	1.47	3.47	3.47	6.29	1.29	1.59	2.61	4.85	1.52	1.93	3.47	6.27
1.60	1.29	1.60	2.60	4.88	1.53	3.47	3.47	6.29	1.29	1.59	2.61	4.85	1.52	1.93	3.47	6.27
1.70	1.29	1.60	2.60	4.88	1.53	3.47	3.47	6.29	1.29	1.59	2.61	4.85	1.52	1.93	3.96	6.27
1.80	1.30	1.60	2.60	4.88	1.53	3.47	3.47	6.29	1.29	1.59	2.61	4.85	1.52	2.06	3.96	7.93
1.90	1.30	1.60	2.60	4.88	1.53	3.47	3.47	6.29	1.29	1.59	2.61	4.85	1.52	2.06	3.96	7.93
2.00	1.30	1.60	2.60	4.88	1.53	3.47	3.47	7.98	1.29	1.59	2.61	4.85	1.52	1.93	3.96	7.93

17.4.5 Performance of the VSI EWMA-*R* Control Chart

Concerning the EWMA-R control chart, similar investigations were performed for determining the minimal *ATS* values achieved using the optimal VSI model. These minimal *ATS* values are summarized in Tables 17.16 and 17.17 for shifts $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$ and $n \in \{3, 5, 7, 9\}$. The values used for h_S are 0.5 and 0.1 time units. For comparison purposes, Tables 17.16 and 17.17 also show the minimal *ATS* of the FSI

Table 17.19 Optimal couples (λ^*, K^*) of the VSI EWMA-R for $n \in \{3, 5\}$, $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n=3$															
	$h_S=0.5$								$h_S=0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.13	2.734	0.13	2.734	0.16	2.767	0.16	2.767	0.16	2.767	0.18	2.782	0.19	2.789	0.22	2.802
0.70	0.09	2.656	0.06	2.547	0.09	2.656	0.09	2.656	0.11	2.701	0.11	2.701	0.13	2.734	0.13	2.734
0.80	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.06	2.547	0.06	2.547	0.06	2.547	0.06	2.547
0.90	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
0.95	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.05	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.10	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.20	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.30	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.40	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.50	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.60	0.06	2.547	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.70	0.21	2.798	0.06	2.547	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.80	0.44	2.791	0.06	2.547	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.90	0.45	2.788	0.06	2.547	0.21	2.798	0.45	2.788	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
2.00	0.45	2.788	0.06	2.547	0.21	2.798	0.58	2.75	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
	$n=5$															
	$h_S=0.5$								$h_S=0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.20	2.827	0.20	2.827	0.12	2.734	0.20	2.827	0.28	2.864	0.29	2.866	0.12	2.734	0.29	2.866
0.70	0.13	2.751	0.13	2.751	0.12	2.734	0.14	2.766	0.18	2.811	0.18	2.811	0.12	2.734	0.20	2.827
0.80	0.08	2.633	0.08	2.633	0.08	2.633	0.08	2.633	0.10	2.691	0.10	2.691	0.10	2.691	0.10	2.691
0.90	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
0.95	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.05	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.10	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.20	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.30	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.40	0.12	2.734	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.50	0.17	2.802	0.17	2.802	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.60	0.26	2.857	0.26	2.857	0.12	2.734	0.26	2.857	0.17	2.802	0.05	2.492	0.05	2.492	0.05	2.492
1.70	0.40	2.877	0.39	2.877	0.12	2.734	0.39	2.877	0.22	2.839	0.05	2.492	0.05	2.492	0.05	2.492
1.80	0.45	2.875	0.45	2.875	0.12	2.734	0.45	2.875	0.22	2.839	0.05	2.492	0.05	2.492	0.05	2.492
1.90	0.45	2.875	0.45	2.875	0.12	2.734	0.45	2.875	0.40	2.877	0.05	2.492	0.05	2.492	0.05	2.492
2.00	0.69	2.839	0.56	2.862	0.12	2.734	0.45	2.875	0.43	2.876	0.05	2.492	0.05	2.492	0.05	2.492

Table 17.20 Optimal couples (λ^*, K^*) of the VSI EWMA-R for $n \in \{7, 9\}$, $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1.05, 1.1, 1.2, \ldots, 2\}$, $h_S \in \{0.1, 0.5\}$, $W = \{0.1, 0.3, 0.6, 0.9\}$, $ATS_0 = 370.4$

	$n = 7$															
	$h_S = 0.5$								$h_S = 0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.27	2.880	0.26	2.876	0.10	2.696	0.25	2.871	0.38	2.907	0.38	2.907	0.10	2.696	0.35	2.902
0.70	0.17	2.812	0.18	2.822	0.10	2.696	0.22	2.854	0.22	2.854	0.24	2.866	0.10	2.696	0.24	2.866
0.80	0.11	2.719	0.11	2.719	0.10	2.696	0.11	2.719	0.11	2.719	0.12	2.740	0.10	2.696	0.15	2.788
0.90	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
0.95	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.05	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.10	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.20	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.30	0.10	2.696	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.40	0.21	2.847	0.21	2.847	0.10	2.696	0.14	2.774	0.14	2.774	0.05	2.492	0.05	2.492	0.05	2.492
1.50	0.30	2.890	0.29	2.887	0.10	2.696	0.21	2.847	0.20	2.840	0.05	2.492	0.05	2.492	0.05	2.492
1.60	0.37	2.905	0.37	2.905	0.10	2.696	0.37	2.905	0.33	2.898	0.05	2.492	0.05	2.492	0.05	2.492
1.70	0.37	2.905	0.37	2.905	0.10	2.696	0.37	2.905	0.37	2.905	0.05	2.492	0.05	2.492	0.05	2.492
1.80	0.53	2.911	0.52	2.911	0.10	2.696	0.37	2.905	0.37	2.905	0.05	2.492	0.05	2.492	0.05	2.492
1.90	0.58	2.909	0.58	2.909	0.10	2.696	0.37	2.905	0.37	2.905	0.05	2.492	0.05	2.492	0.05	2.492
2.00	0.64	2.904	0.62	2.906	0.10	2.696	0.37	2.905	0.52	2.911	0.05	2.492	0.05	2.492	0.37	2.905
	$n = 9$															
	$h_S = 0.5$								$h_S = 0.1$							
	W								W							
	0.9		0.6		0.3		0.15		0.9		0.6		0.3		0.15	
τ	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*	λ^*	K^*
0.60	0.35	2.916	0.35	2.916	0.09	2.670	0.30	2.901	0.41	2.926	0.40	2.925	0.40	2.925	0.34	2.913
0.70	0.21	2.854	0.21	2.854	0.09	2.670	0.21	2.854	0.27	2.889	0.28	2.894	0.28	2.894	0.29	2.898
0.80	0.11	2.722	0.14	2.777	0.09	2.670	0.14	2.777	0.14	2.777	0.15	2.792	0.15	2.792	0.16	2.805
0.90	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.06	2.551	0.06	2.551	0.06	2.551	0.06	2.551
0.95	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.05	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.10	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.20	0.07	2.599	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492	0.05	2.492
1.30	0.13	2.761	0.12	2.743	0.07	2.599	0.07	2.599	0.13	2.761	0.05	2.492	0.05	2.492	0.05	2.492
1.40	0.20	2.846	0.20	2.846	0.09	2.670	0.20	2.846	0.19	2.837	0.05	2.492	0.05	2.492	0.05	2.492
1.50	0.33	2.911	0.33	2.911	0.09	2.670	0.32	2.908	0.32	2.908	0.05	2.492	0.05	2.492	0.05	2.492
1.60	0.34	2.913	0.34	2.913	0.09	2.670	0.34	2.913	0.34	2.913	0.05	2.492	0.05	2.492	0.05	2.492
1.70	0.50	2.933	0.50	2.933	0.09	2.670	0.34	2.913	0.34	2.913	0.05	2.492	0.07	2.599	0.05	2.492
1.80	0.57	2.934	0.57	2.934	0.09	2.670	0.34	2.913	0.50	2.933	0.48	2.932	0.07	2.599	0.34	2.913
1.90	0.63	2.933	0.62	2.933	0.09	2.670	0.34	2.913	0.55	2.934	0.52	2.934	0.07	2.599	0.34	2.913
2.00	0.68	2.931	0.67	2.932	0.09	2.670	0.34	2.913	0.59	2.934	0.05	2.492	0.07	2.599	0.34	2.913

EWMA-R (column $h_L = 1$). Like the VSI EWMA-S^2 control chart, the results clearly indicate that the VSI EWMA-R control chart outperforms the FSI scheme for all considered shifts of variability since the VSI model gives a signal earlier than the FSI model. For example, for $n = 5$, $h_S = 0.5$ and $W = 0.6$ we have $ATS^* = 63.4$ when $\tau = 0.9$, while for the FSI model we have $ATS^* = 75.0$. When $\tau = 1.2$, we have $ATS^* = 16.6$ for the VSI EWMA-R control chart, while for the FSI model we have $ATS^* = 22.5$. Table 17.18 shows the optimal long sampling interval h_L^* for several combinations of n, W, h_S and τ. Finally, Tables 17.19 and 17.20 summarize the optimal couples (λ^*, K^*).

17.5 Conclusions

This chapter presented several EWMA control charts, both with static or adaptive design parameters, as effective means to monitor the process variability, involving both process position and dispersion. The EWMA control charts are a tool of statistical process control widely adopted in the manufacturing environments, due to their sensitivity in the detection of process drifts caused by special causes influencing variability. EWMA charts outperform the statistical performance of traditional Shewhart control charts thanks to the definition of the statistic to be monitored, which contains information about the past process history: this translates into a faster response on the chart to the presence of an out-of-control condition. Reducing the number of samples to be taken between the occurrence of a special cause and its detection on the control chart is very important because this allows the probability of nonconforming units to be controlled. The average run length (*ARL*), defined as the expected number of samples to be taken before a signal from the chart, was assumed as a quantitative parameter to measure the speed of the chart in revealing the occurrence of a special cause. Through this parameter, different control chart schemes can be directly compared assuming as a common constraint the same probability to signal for a false alarm. To evaluate the *ARL* of static EWMAs, two procedures were presented in this chapter: an approach based on the numerical integration of a Fredholm equation and another based on an approximate discrete Markov-chain model. The *ARL* evaluation of the adaptive EWMAs was performed through the Markov chain, which, in this case, allows for an easy mathematical formulation to be modeled. Traditionally, static and adaptive EWMA charts have been implemented in order to monitor the process position with respect to a particular target value: EWMAs for the process mean and median monitoring have been developed in the literature and compared each other or with Shewhart schemes for the sample mean or median. Here, an extensive set of results are presented for the out-of-control *ARL* of these charts: the analysis of the data shows how the static EWMA-$\bar{X}$ always outperforms the static EWMA-$\tilde{X}$ and the corresponding Shewhart scheme for a wide range of assumed drifts, whatever the sample size. Therefore, the adoption of the EWMA-$\bar{X}$ is suggested to the practitioner whenever small process drifts must be detected. However, in statistical quality control the monitoring of

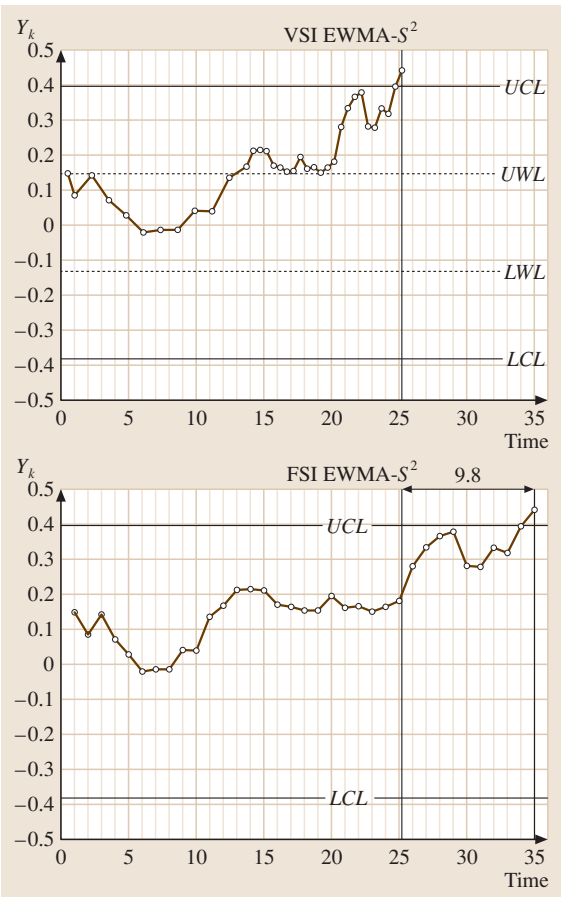


Fig. 17.18 FSI and VSI EWMA-S^2 control chart corresponding to the data in Table 17.15

the process dispersion is equally important as the position, which can be conducted by applying EWMAs to statistics based on the sample dispersion. A survey of EWMAs for monitoring the sample standard deviation S, sample variance S^2 and sample range R was reported in this chapter. The investigated charts work on logarithmic transformations of the measures of dispersion to cope with a variable approximately normally distributed: the formulas adopted to achieve this transformed variable are widely reported throughout the text. The reason for this approach lies in the possibility of managing charts characterized by symmetrical limits, which is common practice in industrial applications to simplify the operator's tasks in the management of the chart. The selected logarithmic transformations work well for all the considered statistics and allow for a direct comparison of the statistical properties of the three charts investigated. The *ARL* computation, optimization and comparison for the three static EWMAs allowed us to demonstrate that the EWMA-S^2 always outperforms both the EWMA-S and EWMA-R when the occurrence of a special cause results in an increase in the dispersion of the data; i.e., when a process is monitored and a deterioration in process repeatability is expected, the EWMA-S^2 is suitable to detect it. Otherwise, if a reduction in process dispersion is expected, for example after a machine maintenance intervention or technological improvement, then the EWMA-R or EWMA-S should be used: in particular, for small sample sizes, ($n \leq 5$), implementing the EWMA monitoring the sample range R is statistically more correct than the EWMA-S, due to the scarce number of measures adopted to evaluate the dispersion statistic. When $n > 5$, the EWMA-S should be used. A further step in the analysis involved the investigation of the adaptive version of the EWMA-S^2 and EWMA-R charts. These adaptive charts have a sampling frequency which is a function of the position of the last plotted point on the chart. The underlying idea is that, when a point is plotted near to the control limits, a possible special cause could be occurring, even if the chart still has not signaled an out-of-control condition; if this is the case, then the next sample should be taken after a shorter time interval. For this reason the variable sampling interval versions of EWMA-S^2 and EWMA-R have been designed and investigated; the EWMA control interval was divided into three zones: a point falling within the central zone, containing the central line of the chart, calls for a longer sampling interval, whereas a point plotted within one of the two external zones included between the central zone and the control limits calls for a shorter time interval. Due to the variability of the time between two samples, the statistical efficiency of these charts was not measured through the *ARL* but through the average time to signal *ATS*. *ATS*s were computed for several sample sizes and expected shifts in process dispersion through the approximate Markov-chain model. For each of the two investigated charts several tables were reported, including the optimal *ATS*s, a comparison with the corresponding static schemes and the values of the optimal parameters. The results show that the variable sampling interval charts always statistically outperform the static charts; furthermore, the differences between S^2 and R found for the static schemes also occur for the adaptive versions: therefore, when increases in process dispersion are expected, the VSI EWMA-S^2 should be used, otherwise the VSI EWMA-R represents the best choice. Finally, it must be argued that EWMA schemes represent a more powerful tool than the traditional Shewhart control charts when process variability is to be monitored, enhancing their statistical properties through the possibility of varying the sampling frequency is effective, whichever the entity of the drift. As a consequence future research should be devoted to the development of adaptive schemes involving the possibility of changing sample size or both sample frequency and sample size.

References

17.1 S. W. Roberts: Control chart tests based on geometric moving averages, Technometrics **1**(3), 239–250 (1959)

17.2 P. B. Robinson, T. Y. Ho: Average run lengths of geometric moving average charts by numerical methods, Technometrics **20**(1), 85–93 (1978)

17.3 S. V. Crowder: A simple method for studying run-length distributions of exponentially weighted moving average charts, Technometrics **29**(4), 401–407 (1987)

17.4 S. V. Crowder: Design of exponentially weighted moving average schemes, J. Qual. Technol. **21**(3), 155–162 (1989)

17.5 J. M. Lucas, M. S. Saccucci: Exponentially weighted moving average control schemes: Properties and enhancements, Technometrics **32**(1), 1–12 (1990)

17.6 S. H. Steiner: Exponentially weighted moving average control charts with time varying control limits and fast initial response, J. Qual. Technol. **31**, 75–86 (1999)

17.7 A. W. Wortham, L. J. Ringer: Control via exponential smoothing, Logist. Rev. **7**(32), 33–40 (1971)

17.8 A. L. Sweet: Control charts using coupled exponentially weighted moving averages, IIE Trans. **18**, 26–33 (1986)

17.9 C. H. Ng, K. E. Case: Development and evaluation of control charts using exponentially weighted moving averages, J. Qual. Technol. **21**(4), 242–250 (1989)

17.10 S. V. Crowder, M. D. Hamilton: An EWMA for monitoring a process standard deviation, J. Qual. Technol. **24**(1), 12–21 (1992)

17.11 M. D. Hamilton, S. V. Crowder: Average run lengths of EWMA control charts for monitoring a process standard deviation, J. Qual. Technol. **24**(1), 44–50 (1992)

17.12 J. F. MacGregor, T. J. Harris: The exponentially weighted moving variance, J. Qual. Technol. **25**(2), 106–118 (1993)

17.13 F. F. Gan: Joint monitoring of process mean and variance using exponentially weighted moving average control charts, Technometrics **37**, 446–453 (1995)

17.14 R. W. Amin, H. Wolff, W. Besenfelder, R. Jr. Baxley: EWMA control charts for the smallest and largest observations, J. Qual. Technol. **31**, 189–206 (1999)

17.15 C. W. Lu, M. R. Jr. Reynolds: Control charts for monitoring the mean and variance of autocorrelated processes, J. Qual. Technol. **31**, 259–274 (1999)

17.16 C. A. Acosta-Mejia, J. J. Jr. Pignatiello, B. V. Rao: A comparison of control charting procedures for monitoring process dispersion, IIE Trans. **31**, 569–579 (1999)

17.17 P. Castagliola: *An EWMA control chart for monitoring the logarithm of the process sample variance, Proc. International Conference on Industrial Engineering and Production Management* (FUCAM, Mons 1999) pp. 371–377

17.18 W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery: *Numerical Recipes in C* (Cambridge Univ. Press, Cambridge 1988)

17.19 D. Brook, D. A. Evans: An approach to the probability distribution of CUSUM run length, Biometrika **59**(3), 539–549 (1972)

17.20 P. Castagliola: An $(\tilde{X}/R)$-EWMA control chart for monitoring the process sample median, Int. J. Reliab. Qual. Safety Eng. **8**(2), 123–125 (2001)

17.21 P. Castagliola: Approximation of the normal sample median distribution using symmetrical Johnson S_U distributions: Application to quality control, Commun. Stat. Simul. Comput. **27**(2), 289–301 (1998)

17.22 G. E. Box, W. G. Hunter, J. S. Hunter: *Statistics for Experimenters* (Wiley, New York 1978)

17.23 N. L. Johnson, S. Kotz, N. Balakrishnan: *Continuous Univariate Distributions* (Wiley, New York 1994)

17.24 P. Castagliola: A New S^2-EWMA Control Chart for Monitoring the Process Variance, Qual. Reliab. Eng. Int. **21**(8) (2005)

17.25 A. Stuart, J. K. Ord: *Kendall's Advanced Theory of Statistics*, Vol. 1 (Edward Arnold, London 1994)

17.26 P. Castagliola: *A New EWMA Control Chart for Monitoring the Process Standard-Deviation, Proc. 6th ISSAT International Conference on Reliability and Quality in Design* (ISSAT, New Brunswick 2000) pp. 233–237

17.27 P. Castagliola: A *R*-EWMA control chart for monitoring the process range, Int. J. Reliab. Qual. Safety Eng. **12**(1), 31–49 (2005)

17.28 M. R. Reynolds Jr, R. W. Amin, J. Arnold, J. Nachlas: $\bar{X}$ charts with variable sampling intervals, Technometrics **30**(2), 181–192 (1988)

17.29 M. R. Reynolds Jr, R. W. Amin, J. Arnold: CUSUM chart with variable sampling intervals, Technometrics **32**(4), 371–384 (1990)

17.30 G. Runger, J. J. Pignatiello Jr: Adaptive sampling for process control, J. Qual. Technol. **23**(2), 135–155 (1991)

17.31 M. S. Saccucci, R. W. Amin, J. Lucas: Exponentially weighted moving average control scheme with variable sampling intervals, Commun. Stat. Simul. Comput. **21**(3), 627–657 (1992)

17.32 M. R. Reynolds Jr: Shewhart and EWMA variable sampling interval control charts with sampling at fixed times, J. Qual. Technol. **28**(2), 199–212 (1996)

17.33 I. Chengular, J. Arnolds, M. R. Reynolds Jr.: Variable sampling intervals for multiparameter Shewhart charts, Commun. Stat. Theory Methods **18**, 1769–1792 (1993)

17.34 M. R. Reynolds Jr., Z. Stoumbos: Monitoring the process mean and variance using individual observations and variable sampling intervals, J. Qual. Technol. **33**(2), 181–205 (2001)

17.35 M. Reynolds: Evaluation properties of variable sampling interval control charts, Sequential Anal. **14**(1), 59–97 (1995)

17.36 P. Castagliola, G. Celano, S. Fichera, F. Giuffrida: A variable sampling interval S^2-EWMA control chart for monitoring the process variance, Int. J. Technol. Manage. (2006) to be published

18. Multivariate Statistical Process Control Schemes for Controlling a Mean

The quality of products produced and services provided can only be improved by examining the process to identify causes of variation. Modern production processes can involve tens to hundreds of variables, and multivariate procedures play an essential role when evaluating their stability and the amount of variation produced by common causes. Our treatment emphasizes the detection of a change in level of a multivariate process.

After a brief introduction, in Sect. 18.1 we review several of the important univariate procedures for detecting a change in level among a sequence of independent random variables. These include Shewhart's X–bar chart, Page's cumulative sum, Crosier's cumulative sum, and exponentially weighted moving-average schemes.

Multivariate schemes are examined in Sect. 18.2. In particular, we consider the multivariate T^2 chart and the related bivariate ellipse format chart, the cumulative sum of T chart, Crosier's multivariate scheme, and multivariate exponentially weighted moving-average schemes.

An application to a sheet metal assembly process is discussed in Sect. 18.3 and the various multivariate procedures are illustrated.

Comparisons are made between the various multivariate quality monitoring schemes in Sect. 18.4. A small simulation study compares average run lengths of the different procedures under some selected persistent shifts.

When the number of variables is large, it is often useful to base the monitoring procedures on principal components. Section 18.5 discussesthis approach. An example is also given using the sheet metal assembly data.

Finally, in Sect. 18.6, we warn against using the standard monitoring procedures without first checking for independence among the observations. Some calculations, involving first-order autoregressive dependence, demonstrate that dependence causes a substantial deviation from the nominal average run length.

Today, with automated data collection a common practice, data on many characteristics need to be continually monitored. In this chapter, we briefly review the major univariate methods and then discuss the multivariate quality monitoring methods for detecting a change in the level of a process. We concentrate on the sequential schemes where the average run length curve is of primary importance for describing the performance. All univariate monitoring schemes attempt to determine if a sequence of observations $X_1, X_2, \cdots$ is stable. That is, to confirm that the mean and variance remain constant. Throughout our discussion we assume that the X_i are in-

dependent. However, in the last section, we do consider the effect of dependence on average run length.

We begin with a review of univariate procedures in Sect. 18.1 and then go on to multivariate extensions in Sect. 18.2. In Sect. 18.3, an example is used to illustrate the behavior of different multivariate procedures. The performance of the multivariate schemes is also compared via simulation in Sect. 18.4. Control charts based on principle components are introduced in Sect. 18.5. In Sect. 18.6, we discuss the difficulties caused by time dependence in the sequence of observations.

18.1 Univariate Quality Monitoring Schemes

To set notation, let $X_1, X_2, \cdots$ be the sequence of independent random variables produced by a process being monitored. Let a denote the target mean value. In the fixed-sample-size setting, the hypothesis to be tested is $H_0 : \mu = a$ versus $H_1 : \mu \neq a$. However, in the sequential setting, we develop statistics based on the deviation from the target value. The statistics will all involve s, an estimate of the standard deviation of a single observation.

Typically, the sequential schemes involve a constant k, called a reference value, and a positive constant h, that defines the decision rule. The two constants (k, h) are selected to give good average run length (ARL) properties. Sometimes, separate reference values k^+ and k^- are used to detect an increase and a decrease in the mean, respectively.

Figure 18.1 shows 50 observations collected on an automotive sheet metal assembly process. These measurements, made by sensors, are deviations from the nominal values (millimeters) at the back right-hand side of the car body. Measurements made at various locations of the car body are presented along with more details in Sect. 18.4.

Figures 18.2–18.5 present different univariate statistics for detecting a change within that data.

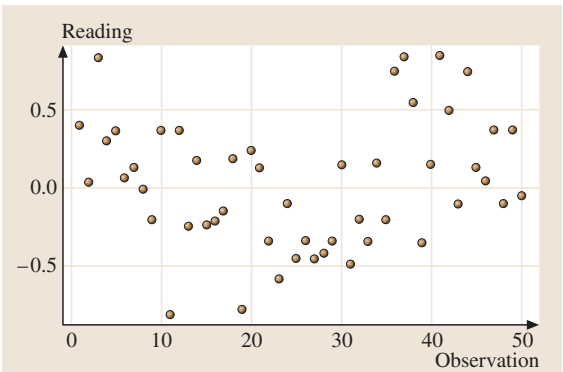


Fig. 18.1 Automotive assembly data

18.1.1 Shewhart *X*-Bar Chart

Historically, the first quality control chart was defined by *Shewhart* [18.1]. This procedure has been widely used since the 1940s. The Shewhart X-bar chart is generally based on the mean of a small sample. That is, the plotted point is usually the mean $\overline{x}_n$ of a few $(m \geq 1)$ observations. When samples are available, rather than just individual observations, charts are also maintained to monitor the process standard deviation.

A widely used Shewhart's X-bar chart [18.1] signals a shift in mean when

$$\overline{X}_n \geq a + 3\frac{s}{\sqrt{m}} \quad \text{or} \quad \overline{X}_n \leq a - 3\frac{s}{\sqrt{m}} ,$$

where s is an estimate of the standard deviation of an individual observation obtained from data collected during stable operation.

Figure 18.2 illustrates Shewhart's X-bar chart applied to automotive assembly data with $m = 1$.

Figure 18.1 suggests that there is a small increase in the mean towards the end of the sequence. However, it is not detected by the Shewhart's X-bar chart.

The Shewhart X-bar chart is very simple and effective for detecting an isolated large shift. However,

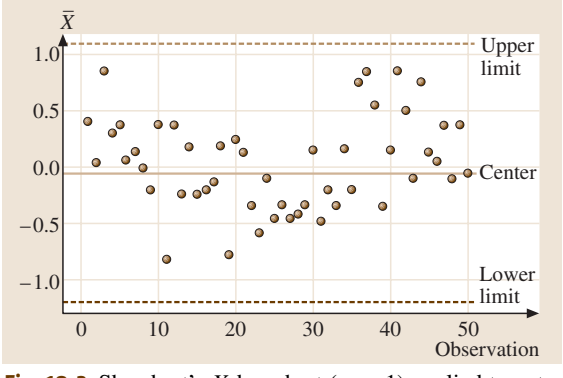


Fig. 18.2 Shewhart's X-bar chart $(m = 1)$ applied to automotive assembly data

because any decision using the Shewhart chart is based only on the most recent observation and has no memory, it is not effective in detecting small or moderate shifts, even if the shifts are persistent.

To help remedy the insensitivity to small shifts, practitioners often apply additional rules for signaling a change. These conditions to signal include:

i) nine points in a row on the same side of the centerline

ii) six points in a row that are decreasing or six that are increasing.

18.1.2 Page's Two-Sided CUSUM Scheme

The univariate two-sided cumulative sum (CUSUM) scheme proposed by *Page* [18.2] uses two cumulative sums: one to detect an increase in mean and another to detect a decrease. It is based on single observations X_n rather than means.

For $n > 1$, Page iteratively defines the two statistics

$$S_{H(n)} = \max(0, S_{H(n-1)} + X_n - k^+) , \tag{18.1}$$

$$S_{L(n)} = \min(0, S_{L(n-1)} + X_n - k^-) \tag{18.2}$$

with specified starting values $S_{H(0)} \geq 0$, $S_{L(0)} \leq 0$. Separate reference values k^+ and k^- are used in Page's CUSUM scheme to detect an increase in the mean and a decrease in the mean, respectively. Specifically, k^+ is selected to be larger than the target a so each term in the sum has a slightly negative expected value, and k^- is selected to be smaller than a. When the sequence of random variables X_1, $X_2, \cdots$ are normally distributed, typically we can set $k^+ = a + ks$ and $k^- = a - ks$, where k is one half of the specified shift in mean (expressed in standard deviations) that should be quickly detected by the scheme.

If the process remains in control at the target value, the CUSUM statistics defined in (18.1) and (18.2) should vary randomly but stay close to zero. When there is an increase in the process mean, a positive drift will develop in the CUSUM statistics S_H. Conversely, if there is a decrease in the mean, then a negative drift will develop in S_L. Therefore, an increasing trend in S_H or a decreasing trend in S_L is taken to indicate a shift in the process mean.

Page's CUSUM scheme signals a shift in mean when

$S_{H(n)} \geq hs$ (signals an increase), or

$S_{L(n)} \leq -hs$ (signals a decrease) .

The positive constant h is chosen to obtain a desired value of in-control ARL.

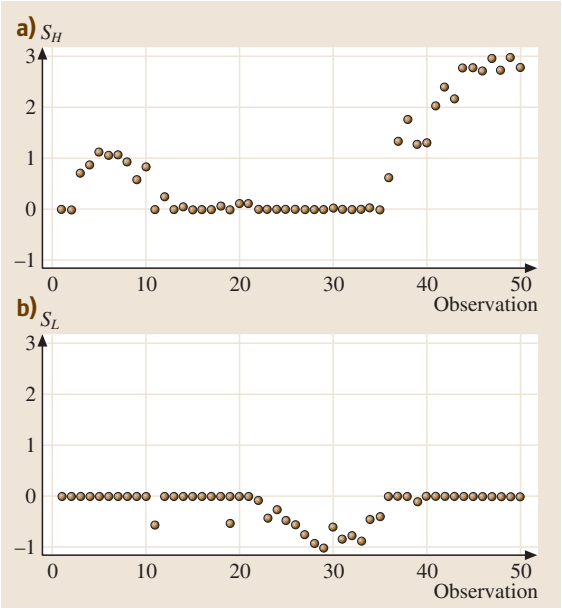


Fig. 18.3 Page's CUSUM statistics S_H (a) and S_L (b) using automotive assembly data

Figure 18.3 illustrates Page's CUSUM statistics applied to the automotive assembly data. The increasing trend in the plot of S_H indicates an increase in the mean towards the end of the sequence. Generally, Page's CUSUM scheme is more effective in detecting small but persistent shifts than the Shewhart X-bar chart.

18.1.3 Crosier's Two-Sided CUSUM Scheme

Crosier [18.3] proposed a two-sided CUSUM scheme which first updates the previous CUSUM by a new observation. Depending on the updated value of this sum, the new value of the CUSUM is either set equal to zero or the sum is shrunk towards zero. This modification reduces the chance of giving a false alarm. It seems from *Crosier* [18.3] that this procedure was arrived at empirically by trying a great many different schemes.

In particular, Crosier's two-sided CUSUM starts with $S_0 = 0$. For each step, first calculate the tentative sum $C_n = |S_{n-1} + (X_n - a)|$. Then Crosier's CUSUM statistic S_n is iteratively defined as

$$S_n = \begin{cases} 0 & \text{if } C_n \leq ks \\ (S_{n-1} + X_n - a)(1 - ks/C_n) & \text{otherwise.} \end{cases} \tag{18.3}$$

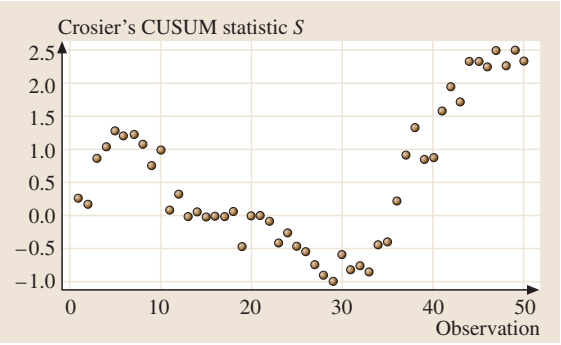


Fig. 18.4 Crosier's CUSUM statistic using automotive assembly data

Here, the constant k can also be set equal to one half of a specified mean-shift (expressed in standard deviations) that should be detected quickly.

Crosier's scheme signals that the mean has shifted when

$$S_n \geq hs \quad \text{(increase)}$$

or

$$S_n \leq -hs \ \text{(decrease)} .$$

Figure 18.4 illustrates Crosier's two-sided CUSUM statistic applied to automotive assembly data.

The plot indicates a possible small decrease in the mean near the middle of the sequence and an increase towards the end.

18.1.4 EWMA Scheme

The univariate exponentially weighted moving-average (EWMA) scheme [18.4] is based on the weighted average of the current CUSUM and the new observation. The EWMA scheme smooths the sequence of observations by taking an average where the most recent observation receives the highest weight. Starting at $Z_0 = 0$, the updated EWMA sum is defined as

$$Z_n = r(X_n - a) + (1-r)Z_{n-1} \qquad (18.4)$$

for $n = 1, 2, \cdots$, where $0 \leq r \leq 1$ is a specified constant. Expressed in terms of all the observations, we have

$$\begin{aligned} Z_n &= r\sum_{i=0}^{n-1}(1-r)^i(X_{n-i} - a) \\ &= r(X_n - a) + r(1-r)(X_{n-1} - a) \\ &\quad + r(1-r)^2(X_{n-2} - a) + \cdots , \end{aligned}$$

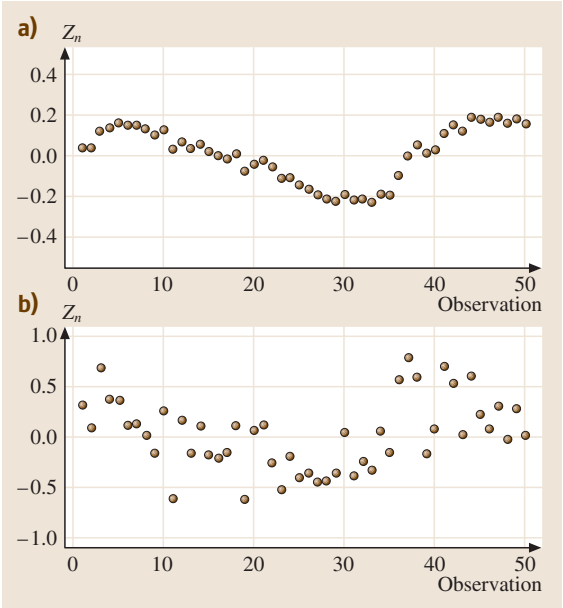


Fig. 18.5 EWMA with $r = 0.1$ (a) and $r = 0.8$ (b), using automotive assembly data

where the weights $r(1-r)^i$ fall off exponentially, giving rise to the name of the EWMA scheme.

The EWMA scheme signals a change in mean when

$$Z_n \geq hs \ \text{(increase)} \ \text{ or } \ Z_n \leq -hs \ \text{(decrease)} .$$

Figure 18.5 shows the EWMA statistics with different choices of r applied to automotive assembly data. In the plot we can see that the EWMA statistic with the smaller value of parameter r is more sensitive to small shifts in this process mean. To be specific, the plot of the EWMA statistics with $r = 0.1$ has a decreasing trend in the middle and then an increasing towards the end of the sequence, which suggests a decrease and then an increase in the mean. However, in the plot of EWMA statistics with $r = 0.8$, such patterns can hardly be recognized.

As shown in Fig. 18.5, the choice of the value for r is critical to the performance of an EWMA scheme. Usually a small value for r is used for detecting a small shift. *Lucas* and *Saccucci* [18.4] extensively discuss the design of EWMA control schemes. For normal observations, they provide a table of *optimal* parameters r and h, which give the minimum ARL at the specified shift in process for specified in-control ARL. Lucas and Saccucci also compare the ARLs of the EWMA and Page's CUSUM schemes. They show that, with carefully chosen parameters r and h, the ARLs for EWMA are usually

smaller than the ARLs of Page's CUSUM scheme when the shift is smaller than the specified value for the shift that the scheme is designed to detect. However, the ARLs for the EWMA are larger than those of Page's CUSUM procedure when the shift in process mean is larger than the specified value, unless r is very large.

18.1.5 Summary Comments

Our review above included most of the popular univariate quality monitoring schemes. See [18.5] for further discussion of univariate schemes for process control. The well-known Shewhart X-bar chart is very effective for detecting large shifts but may be slow to detect small or moderate shifts. This fact gave rise to the setting of warning limits and rules concerning the number of consecutive observations that are increasing or decreasing in value. Page's CUSUM scheme uses a positive (negative) drift when defining the CUSUM statistics for detecting an increase (decrease) in mean. Crosier's CUSUM scheme shrinks the CUSUM statistic instead of adding a drift. Both of these CUSUM procedures make good choices for detecting small but persisting shifts in the process mean. They provide similar ARL performance according to our simulation study in Sect. 18.5 and in *Li* [18.6]. The EWMA scheme proposed by *Lucas* and *Saccucci* [18.4] is also an effective method for detecting small and persisting shifts. Unlike the other two univariate CUSUM schemes, the EWMA does not require the user to specify the shift in mean that should be detected. The EWMA scheme could have better ARL performance than these other CUSUM schemes when the shift is smaller than the specified value. However, the choice of the weight parameter r in (18.4) is somewhat crucial.

18.2 Multivariate Quality Monitoring Schemes

Often, more than one quality characteristics is measured on each unit. We then model the observations as a sequence of independent $p \times 1$ random vectors $\boldsymbol{X}_1$, $\cdots$, $\boldsymbol{X}_n$, $\cdots$ where each has the target mean value $\boldsymbol{a}$ and the same covariance matrix $\boldsymbol{\Sigma}$. If $\boldsymbol{\Sigma}$ is unknown, it must be estimated from a long sequence of observations taken when the process is stable. Any out-of-control observations, detected by a T^2 chart, are deleted. The sample covariance matrix is calculated from the reduced data set. Usually only one data cleaning stage is conducted. As in most of the literature, we will describe the multivariate monitoring schemes in terms of known covariance matrix $\boldsymbol{\Sigma}$. Most of the multivariate schemes discussed below involve a constant k and a positive control limit h.

The literature on multivariate process control includes [18.7–11]. Further discussion is given in [18.12], [18.13, Chapt. 10], and [18.14, Sects. 5.6 and 8.6].

To illustrate what might be the typical behavior of the various multivariate statistics, we generated a sequence of 100 bivariate normal observations with $\boldsymbol{\Sigma} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$. Our process mean shifted from (0,0) to (0,1) after the 40th observation. This is a moderate shift of one standard deviation in one of the two components.

In this section, we apply each monitoring scheme to that single bivariate sequence of observations. In Sect. 18.5, we look at various other choices and perform simulation studies to determine the ARL under a range of alternatives.

18.2.1 Multivariate T^2 Chart

The traditional T^2 chart reduces each multivariate observation to a scalar by defining

$$T_n^2 = (\boldsymbol{X}_n - \boldsymbol{a})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{X}_n - \boldsymbol{a}) \,. \tag{18.5}$$

The multivariate T^2 scheme signals a shift in mean when T_n^2 first exceeds a specified level h. That is, a change in mean is signaled when

$$T_n^2 \geq h \,.$$

The usual upper control limit is the upper χ^2 percentile $\chi_p^2(\alpha)$ with $\alpha = 0.01$. If the estimated covariance matrix $\boldsymbol{S}$ is based on a relatively small number of observations K, then the appropriate upper control limit is

$$\mathrm{UCL} = \frac{p(K-1)}{K-p} F_{p,K-p}(\alpha) \,.$$

As in the case with its univariate counterpart, the Shewhart X-bar chart, the multivariate T^2 procedure is not very sensitive to small or moderate shifts from the target $\boldsymbol{a}$ because it is only based on the most recent observation and has no *memory* of previous observations.

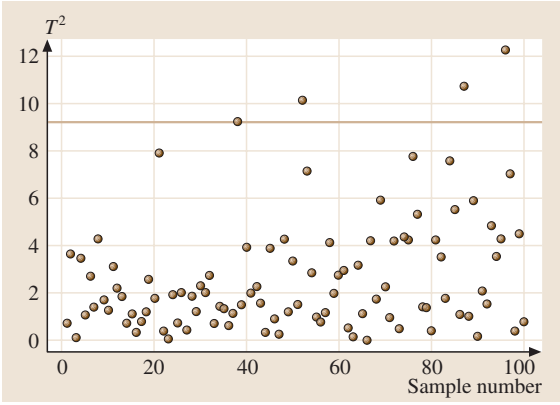


Fig. 18.6 T^2 chart with a 99% control limit using our generated bivariate data

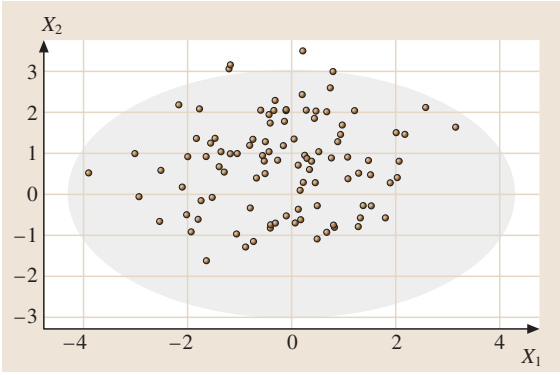


Fig. 18.7 The T^2 99% control ellipse using our generated bivariate data

Figure 18.6 illustrates the traditional multivariate T^2 scheme applied to our generated bivariate normal observations. As shown in the graph, there is a false alarm at $n = 38$ and the first correct detection of a shift is at $n = 52$. The shift from (0,0) to (1,0) at $n = 40$ is not detected particularly quickly and persistently by the T^2 chart with $\boldsymbol{a} = (0, 0)$.

When there are only two important variables, it is more informative to plot the individual 2×1 vectors $\boldsymbol{X}_n$ with the control ellipse in an ellipse format chart for T^2 statistics. The 99% control ellipse is

$$T_n^2 = (\boldsymbol{X}_n - \boldsymbol{a})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{X}_n - \boldsymbol{a}) \le \chi_2^2(0.01) \, .$$

The generated bivariate normal data and the 99% control ellipse are shown in Fig. 18.7. There is one false detection outside the control ellipse at $n = 38$ and a few correct detections of the shift in mean at $n = 52$, 87 and 96.

T^2 Control Charts Based on Means

Often, rather than using a single observation, a control chart is based on points that correspond to the mean of a small sample of size m. We still assume that the population is $p-$variate normal with mean vector $\boldsymbol{0}$ and covariance matrix $\boldsymbol{\Sigma}$. However, because of the central limit effect that sample means are more normal than individual observations, the normality of the population is not as crucial. It is now assumed that each random vector of observations from the process is independently distributed as $N_p(\boldsymbol{0}, \boldsymbol{\Sigma})$. We proceed differently, when the sampling procedure specifies that $m > 1$ units be selected, at the same time, from the process. From the first sample, we determine its sample mean $\overline{\boldsymbol{X}}_1$ and covariance matrix $\mathbf{S}_1$. When the population is normal, these two random quantities are independent.

Starting with K samples, where the the $j-$th sample has mean vector $\overline{\boldsymbol{X}}_j$ and covariance matrix $\mathbf{S}_j$, the estimator of the population mean vector $\boldsymbol{\mu}$ is the overall sample mean

$$\overline{\overline{\boldsymbol{X}}} = \frac{1}{K} \sum_{j=1}^{K} \overline{\boldsymbol{X}}_j \, .$$

The sample covariances from the n samples can be combined to give a single estimate (called $\boldsymbol{S}_{\text{pooled}}$) of $\boldsymbol{\Sigma}$ as

$$\mathbf{S} = \frac{1}{K} (\mathbf{S}_1 + \mathbf{S}_2 + \cdots + \mathbf{S}_K) \, ,$$

where $(mK - K)\mathbf{S}$ is independent of each $\overline{\boldsymbol{X}}_j$ and therefore of their mean $\overline{\overline{\boldsymbol{X}}}$. That is, we can now estimate $\boldsymbol{\Sigma}$ internally from the data collected in any given period. These estimators are combined into a single estimator with a large number of degrees of freedom.

T^2 Chart When Means Are Plotted. When the chart is based on the sample mean $\overline{\boldsymbol{X}}_n$ of m observations rather than a single observation, the values of

$$T_n^2 = m(\overline{\boldsymbol{X}}_n - \boldsymbol{a})' \boldsymbol{\Sigma}^{-1} (\overline{\boldsymbol{X}}_n - \boldsymbol{a})$$

are plotted for $n = 1, 2, \cdots$, where the

$$\text{UCL} = \chi_p^2(0.01)$$

or some other upper percentile of the chi-square distribution with p degrees of freedom.

Ellipse Format Chart. The control ellipse, expressed in terms of the sample mean $\overline{\boldsymbol{X}}_n$ of m observations, is

$$(\overline{\boldsymbol{x}} - \boldsymbol{a})' \boldsymbol{\Sigma}^{-1} (\overline{\boldsymbol{x}} - \boldsymbol{a}) \le \chi_p^2(0.01)/m \, ,$$

where $\chi^2_p(0.01)/m$, on the right-hand side, can be replaced by some other upper percentile. For $p = 2$, a graph of the ellipse is usually presented for visual inspection of the data.

18.2.2 CUSUM of T_n (COT) Scheme

The cumulative sum of T, COT or CUSUM of T_n scheme, is the most direct extension of the multivariate T^2 chart to a CUSUM procedure. It forms a CUSUM of the scalar statistics T_n, where T_n^2 is defined in (18.5).

Let $S_0 \geq 0$ and $k > 0$ be specified constants. Iteratively define

$$S_n = \max(0, S_{n-1} + T_n - k), \quad \text{for} \quad n = 1, 2, \cdots, \tag{18.6}$$

where k is a specified positive constant.

The COT scheme signals when it surpasses a specified level h. That is, the COT scheme signals a change in mean when

$$S_n \geq h \, .$$

Figure 18.8 illustrates the performance of the CUSUM of T. As we can see, the COT scheme does show an increasing trend which indicates a shift in the sequence mean, while the multivariate T^2 chart does not persistently indicate a change because the shift is not large enough.

18.2.3 Crosier's Multivariate CUSUM Scheme

Crosier [18.11] also generalized his univariate CUSUM scheme to the multivariate setting. Crosier's multivariate statistic starts at $\boldsymbol{S}_0 = \boldsymbol{0}$. With a specified constant k, iteratively define

$$\boldsymbol{S}_n = \begin{cases} \boldsymbol{0} & \text{if } C_n \leq k \\ (\boldsymbol{S}_{n-1} + \boldsymbol{X}_n - \boldsymbol{a})(1 - k/C_n) & \text{otherwise,} \end{cases} \tag{18.7}$$

where

$$C_n = [(\boldsymbol{S}_{n-1} + \boldsymbol{X}_n - \boldsymbol{a})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{S}_{n-1} + \boldsymbol{X}_n - \boldsymbol{a})]^{1/2} \, .$$

For a specified constant h, Crosier's multivariate scheme signals a shift in mean from $\boldsymbol{a}$ when

$$Y_n = (\boldsymbol{S}_n' \boldsymbol{\Sigma}^{-1} \boldsymbol{S}_n)^{1/2} \geq h \, .$$

Figure 18.9 illustrates the performance of Crosier's multivariate CUSUM scheme with the generated bivariate normal data. In the plot, a increasing trend was shown shortly after the shift in process occurs at the 41st observation. Therefore, the shift is detected faster than it is detected by the COT scheme.

18.2.4 Multivariate EWMA Scheme [MEWMA(r)]

In multivariate settings, *Lowry* and *Woodall* [18.15] proposed a natural extension of the univariate EWMA scheme. Starting with $\boldsymbol{Z}_0 = \boldsymbol{0}$, the multivariate EWMA statistic $\boldsymbol{Z}_n$ is defined, iteratively, by

$$\boldsymbol{Z}_n = \mathbf{R}(\boldsymbol{X}_n - \boldsymbol{a}) + (\mathbf{I} - \mathbf{R})\boldsymbol{Z}_{n-1} \quad \text{for} \quad n = 1, 2, \cdots, \tag{18.8}$$

where the weight matrix $\mathbf{R} = diag(r_1, \cdots, r_p), 0 \leq r_j \leq 1, \, j = 1, \cdots, p$. That is, Lowry and Woodall specialize

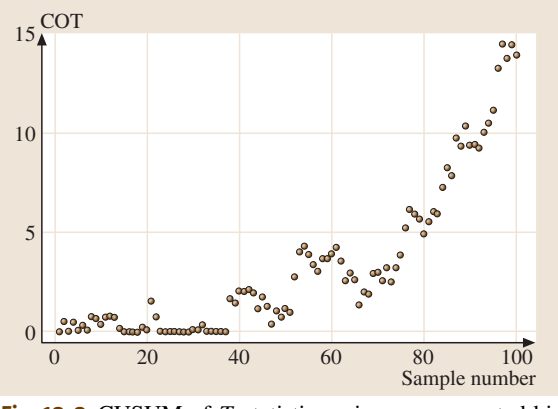


Fig. 18.8 CUSUM of T statistics using our generated bivariate data

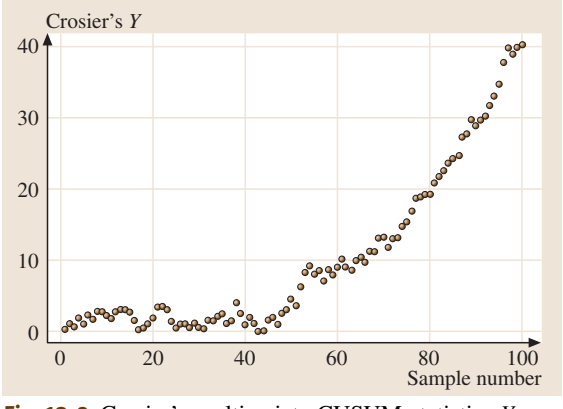


Fig. 18.9 Crosier's multivariate CUSUM statistics Y_n applied to our generated normal data

Index	x_1	x_2	x_3	x_4	x_5	x_6
1	−0.12	0.36	0.40	0.25	1.37	−0.13
2	−0.60	−0.35	0.04	−0.28	−0.25	−0.15
3	−0.13	0.05	0.84	0.61	1.45	0.25
4	−0.46	−0.37	0.30	0.00	−0.12	−0.25
5	−0.46	−0.24	0.37	0.13	0.78	−0.15
6	−0.46	−0.16	0.07	0.10	1.15	−0.18
7	−0.46	−0.24	0.13	0.02	0.26	−0.20
8	−0.13	0.05	−0.01	0.09	−0.15	−0.18
9	−0.31	−0.16	−0.20	0.23	0.65	0.15
10	−0.37	−0.24	0.37	0.21	1.15	0.05
11	−1.08	−0.83	−0.81	0.05	0.21	0.00
12	−0.42	−0.30	0.37	−0.58	0.00	−0.45
13	−0.31	0.10	−0.24	0.24	0.65	0.35
14	−0.14	0.06	0.18	−0.50	1.25	0.05
15	−0.61	−0.35	−0.24	0.75	0.15	−0.20
16	−0.61	−0.30	−0.20	−0.21	−0.50	−0.25
17	−0.84	−0.35	−0.14	−0.22	1.65	−0.05
18	−0.96	−0.85	0.19	−0.18	1.00	−0.08
19	−0.90	−0.34	−0.78	−0.15	0.25	0.25
20	−0.46	0.36	0.24	−0.58	0.15	0.25
21	−0.90	−0.59	0.13	0.13	0.60	−0.08
22	−0.61	−0.50	−0.34	−0.58	0.95	−0.08
23	−0.61	−0.20	−0.58	−0.20	1.10	0.00
24	−0.46	−0.30	−0.10	−0.10	0.75	−0.10
25	−0.60	−0.35	−0.45	0.37	1.18	−0.30
26	−0.60	−0.36	−0.34	−0.11	1.68	−0.32
27	−0.31	0.35	−0.45	−0.10	1.00	−0.25
28	−0.60	−0.25	−0.42	0.28	0.75	0.10
29	−0.31	0.25	−0.34	−0.24	0.65	0.10
30	−0.36	−0.16	0.15	−0.38	1.18	−0.10
31	−0.40	−0.12	−0.48	−0.34	0.30	−0.20
32	−0.60	−0.40	−0.20	0.32	0.50	0.10
33	−0.47	−0.16	−0.34	−0.31	0.85	0.60
34	−0.46	−0.18	0.16	0.01	0.60	0.35
35	−0.44	−0.12	−0.20	−0.48	1.40	0.10
36	−0.90	−0.40	0.75	−0.31	0.60	−0.10
37	−0.50	−0.35	0.84	−0.52	0.35	−0.75
38	−0.38	0.08	0.55	−0.15	0.80	−0.10
39	−0.60	−0.35	−0.35	−0.34	0.60	0.85
40	0.11	0.24	0.15	0.40	0.00	−0.10
41	0.05	0.12	0.85	0.55	1.65	−0.10
42	−0.85	−0.65	0.50	0.35	0.80	−0.21
43	−0.37	−0.10	−0.10	−0.58	1.85	−0.11
44	−0.11	0.24	0.75	−0.10	0.65	−0.10
45	−0.60	−0.24	0.13	0.84	0.85	0.15
46	−0.84	−0.59	0.05	0.61	1.00	0.20
47	−0.46	−0.16	0.37	−0.15	0.68	0.25
48	−0.56	−0.35	−0.10	0.75	0.45	0.20
49	−0.56	−0.16	0.37	−0.25	1.05	0.15
50	−0.25	−0.12	−0.05	−0.20	1.21	0.10

to cases where $\mathbf{R}$ is a diagonal matrix. This reduces to the situation where a univariate EWMA is applied to each individual component. When there is no a priori reason to weight the p quality characteristics differently, they further suggest the use of a common value $r_1 = \cdots = r_p = r$, where $0 \le r \le 1$ is a constant.

The MEWMA scheme signals if the scalar $\boldsymbol{Z}_n' \boldsymbol{\Sigma}^{-1} \boldsymbol{Z}_n$ is large. More particularly, for a specified constant h, the MEWMA scheme signals a shift in mean from $\boldsymbol{a}$ when

$$(\boldsymbol{Z}_n' \boldsymbol{\Sigma}^{-1} \boldsymbol{Z}_n)^{1/2} \ge h \, .$$

Analogous to the univariate case, the choice of weight matrix $\mathbf{R}$ has a considerable influence on the resulting ARL behavior. Usually a small value of r is selected to detect small shifts in each component of mean.

Figure 18.10 illustrates the performance of the MEWMA scheme evaluated for our generated bivariate normal data under two different choices of the constant r. Again, we see that the MEWMA scheme with a small value of r is more effective in detecting a small but persisting shift like the one in our generated data.

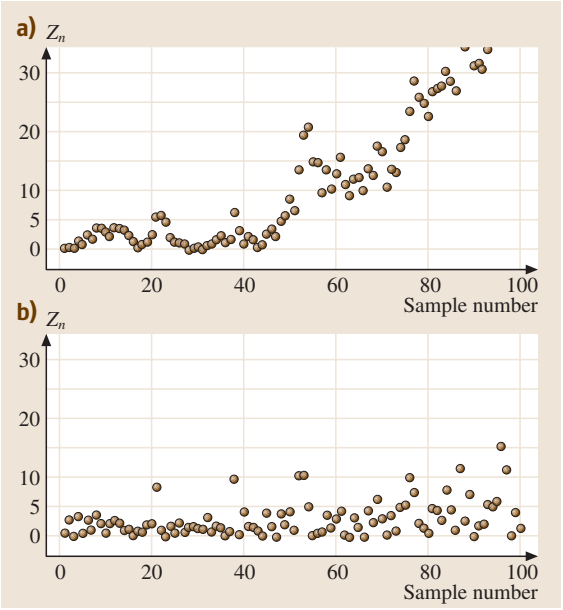


Fig. 18.10 MEWMA with $r = 0.1$ (a) and $r = 0.8$ (b), using our generated bivariate normal data

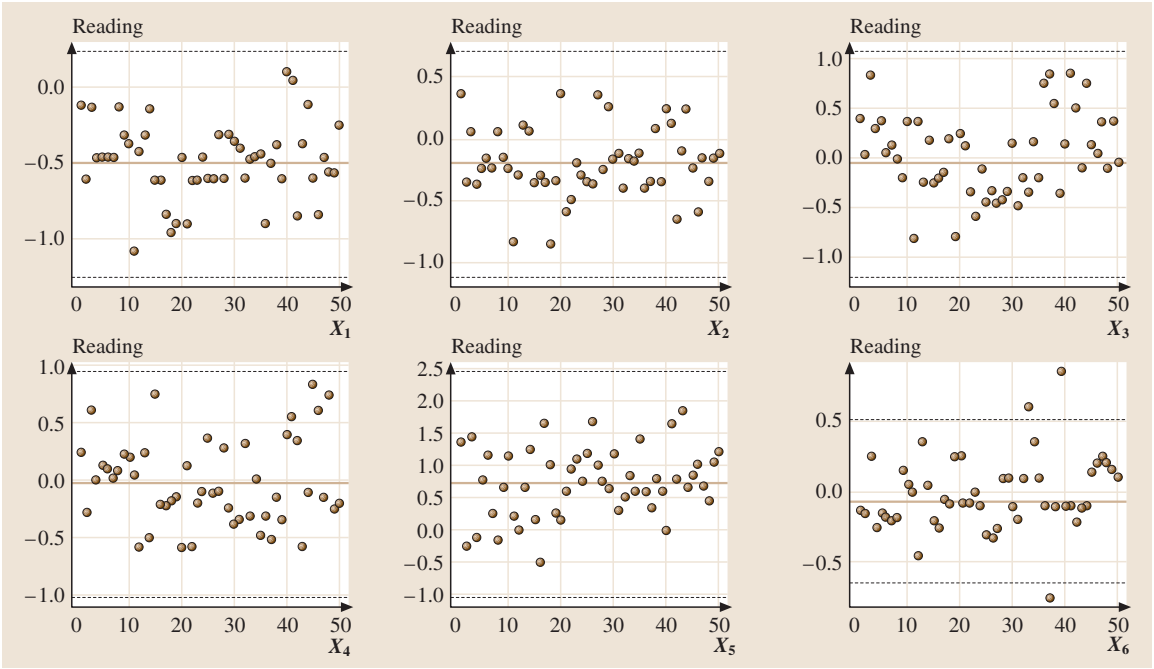


Fig. 18.11 Shewhart $X-$bar chart for each variable in automotive assembly data

18.3 An Application of the Multivariate Procedures

The data that we use to illustrate the various control charts is courtesy of Darek Ceglarek. He collected these measurements on the sheet metal assembly process as part of a study conducted with a major automobile manufacturer. *Ceglarek* and *Shi* [18.16] give more detail on the body assembly process. There are six variables of which four were measured when the car body was complete and two were measured on the underbody at an earlier stage of assembly.

All measurements were taken by sensors that recorded the deviation from the nominal value in millimeters:

x_1 = deviation at mid right-hand side, body complete,
x_2 = deviation at mid left-hand side, body complete,
x_3 = deviation at back right-hand side, body complete,
x_4 = deviation at back left-hand side, body complete,
x_5 = deviation at mid right-hand side of underbody,
x_6 = deviation at mid left-hand side of underbody.

The covariance matrix and the mean estimated from the first 30 observations are

$$\mathbf{S} = \begin{pmatrix} 0.0626 & 0.0616 & 0.0474 & 0.0083 & 0.0197 & 0.0031 \\ 0.0616 & 0.0924 & 0.0268 & -0.0008 & 0.0228 & 0.0155 \\ 0.0474 & 0.0268 & 0.1446 & 0.0078 & 0.0211 & -0.0049 \\ 0.0083 & -0.0008 & 0.0078 & 0.1086 & 0.0221 & 0.0066 \\ 0.0197 & 0.0228 & 0.0211 & 0.0221 & 0.3428 & 0.0146 \\ 0.0031 & 0.0155 & -0.0049 & 0.0066 & 0.0146 & 0.0366 \end{pmatrix}$$

$$\bar{\boldsymbol{x}} = (\,-0.5063 \quad -0.2070 \quad -0.0620 \quad -0.0317 \quad 0.6980 \quad -0.0650\,)'.$$

Figure 18.11 gives the Shewhart X-bar charts for each of the six variables. Except for two cases with the last variable measured at the left side of the underbody, all of the observations are within their 99% control limits.

Figures 18.12 and 18.13 show the multivariate T^2 chart and the CUSUM of T chart applied to the automotive assembly data. In the multivariate T^2 chart we can see that a few values between the 30th and 40th observations are out of the control limits. The CUSUM of T chart indicates a small and persistent shift at the end of the sequence, which is not detected by the multivariate T^2 chart.

The Crosier's CUSUM statistic illustrated in Fig. 18.14 also indicates a small and persistent shift, which is consistent with the CUSUM of T chart.

The multivariate EWMA schemes are illustrated in Fig. 18.15. There is a considerable difference in the appearance depending on the choice of the common weight r. In the plot of multivariate EWMA

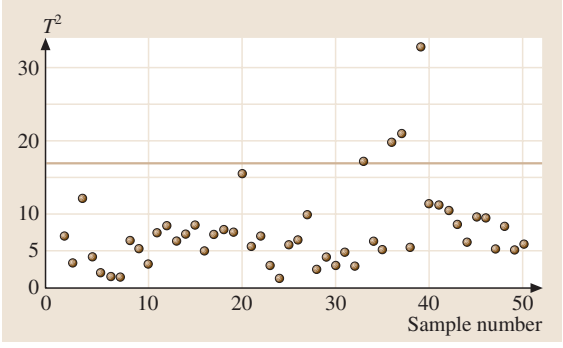


Fig. 18.12 Multivariate T^2 chart for automotive assembly data

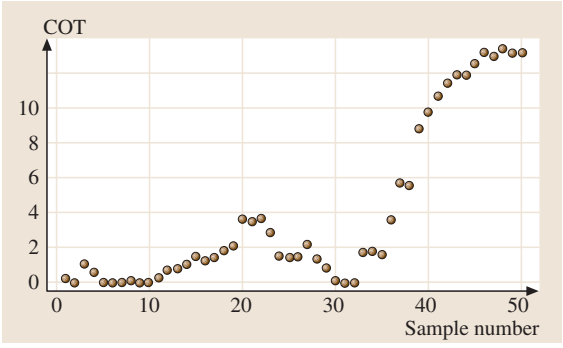


Fig. 18.13 CUSUM of T chart for automotive assembly data

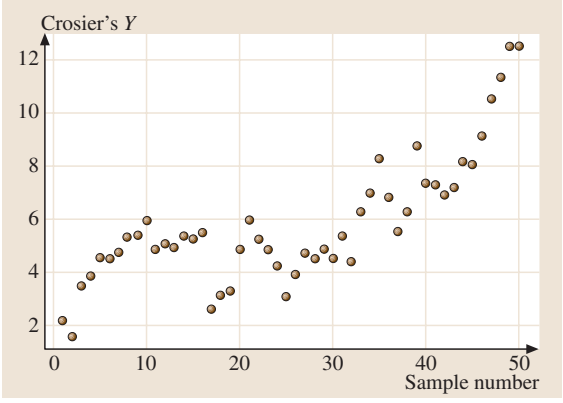


Fig. 18.14 Crosier's CUSUM statistic applied to automotive assembly data

statistics with $r = 0.1$, an increasing trend indicates an increase in the mean, which is consistent with the CUSUM of T chart and Crosier's CUSUM scheme. However, with $r = 0.8$, we can not see the increasing pattern in the plot for the multivariate EWMA statistics.

18.4 Comparison of Multivariate Quality Monitoring Methods

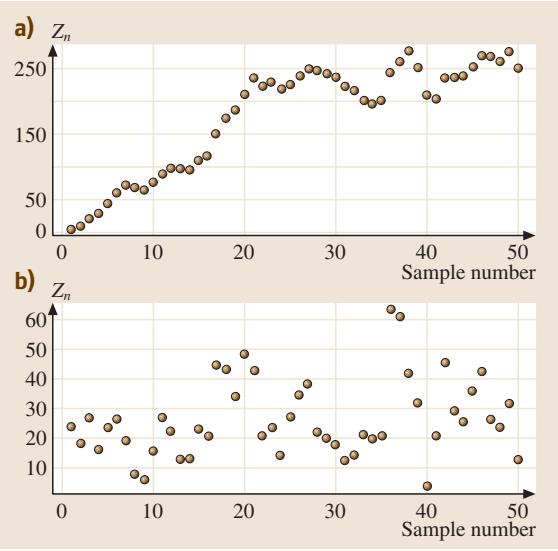


Fig. 18.15a,b Multivariate EWMA statistics with $r = 0.1$ (**a**) and $r = 0.8$ (**b**), using automotive assembly data

Several authors, including *Pignatiello* [18.17] and *Lowry* [18.18], have compared various multivariate monitoring procedures. To confirm their general conclusions, we use simulation to study the average run length (ARL) properties of the multivariate sequential schemes discussed above. Specifically, the ARL is defined as the average number of observations before the scheme gives a signal. If the shift did not occur at the beginning of the series of observations, it is common practice to use the steady-state ARL, which is the average additional run length after the shift occurs. An effective scheme should have a large ARL when the process is *in control* and a small ARL when the process is *out of control*. Usually, to compare different schemes, we can set the in-control ARLs to be nearly equal and compare the ARLs when there is a shift.

In our simulation, we generate bivariate normal observations with either $\boldsymbol{\Sigma} = \boldsymbol{I}$ (uncorrelated) or $\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & -0.6 \\ -0.6 & 1.0 \end{pmatrix}$. For each choice of $\boldsymbol{\Sigma}$ and

Table 18.1 ARL comparison with bivariate normal data (uncorrelated)

Shift	T^2	COT	Crosier	EW(0.1)	EW(0.4)	EW(0.8)
(0,0)	198.3	200.1	200.8	200.4	199.4	198.1
	2.8	2.7	2.9	2.8	2.8	2.8
(1,0)	44.3	20.5	9.5	9.8	13.3	29.6
	0.6	0.2	0.1	0.1	0.1	0.4
(0,-1)	42.9	20.0	9.5	9.8	13.2	28.7
	0.6	0.2	0.1	0.1	0.2	0.4
(0.7,0.7)	42.4	20.3	9.5	9.7	12.7	29.0
	0.6	0.2	0.1	0.1	0.2	0.4
(0.7,-0.7)	42.3	20.3	9.3	9.7	13.4	29.1
	0.6	0.2	0.1	0.1	0.2	0.4
(0.5,0)	119.7	82.4	29.0	27.3	54.2	96.4
	1.6	1.1	0.3	0.3	0.7	1.3
(0.4,-0.4)	113.9	82.1	29.0	27.5	52.8	90.7
	1.6	1.1	0.3	0.3	0.7	1.3
(2,0)	6.9	4.7	4.0	4.3	3.5	4.8
	0.1	0.0	0.0	0.0	0.0	0.1
(1.4,-1.4)	6.7	4.7	4.0	4.3	3.5	4.7
	0.1	0.0	0.0	0.0	0.0	0.1

Table 18.2 ARL comparison with bivariate normal data (correlated)

Shift	T^2	COT	Crosier	EW(0.1)	EW(0.4)	EW(0.8)
(0,0)	201.1	206.5	202.3	203.0	202.7	202.6
	2.8	2.7	3.0	2.8	2.8	2.8
(0.8,0)	42.7	20.0	9.4	9.8	13.1	28.3
	0.6	0.2	0.1	0.1	0.2	0.4
(0,−0.8)	44.6	20.6	9.7	9.9	13.4	29.8
	0.6	0.2	0.1	0.1	0.2	0.4
(0.4,0.4)	43.3	20.0	9.5	9.8	13.1	28.8
	0.6	0.2	0.1	0.1	0.2	0.4
(0.4,−0.4)	115.3	81.8	28.5	26.9	54.2	97.2
	1.6	1.1	0.3	0.3	0.7	1.3
(0.4,0)	114.3	84.1	28.8	27.0	52.8	95.0
	1.6	1.1	0.3	0.3	0.7	1.4
(0.2,−0.2)	172.3	158.2	84.2	71.7	115.5	158.7
	2.4	2.1	1.1	1.0	1.7	2.2
(1.6,0)	7.0	4.7	4.0	4.3	3.5	4.7
	0.1	0.0	0.0	0.0	0.0	0.1
(0.9,−0.9)	41.8	20.4	9.5	10.1	14.4	28.5
	0.6	0.2	0.1	0.1	0.2	0.4

shift in mean, series of observations were generated and the multivariate statistics were calculated until a shift is signaled. This procedure was repeated 5000 times and we calculate the ARL and the estimated standard error of ARL for each scheme. Table 18.1 and 18.2 show the results of our simulation, where the estimated standard errors are given in smaller type.

From our simulation and existing literature, we conclude that, due to the fact that the value of the T^2 statistic only depends on the most current observation, it is not sensitive to small and moderate shifts in the mean of a process even if the shift is persistent. By taking the CUSUM of T, The COT procedure has ARL performance that is significantly improved over that of the T^2 chart. The ARL of Crosier's scheme is considerably better than that of the COT scheme. The performance of multivariate EWMA schemes depend heavily on the value of the weight parameters. If the weight is appropriately selected, the multivariate EWMA will have very good ARL performance which is comparable with Crosier's scheme.

18.5 Control Charts Based on Principal Components

The first two sample principal components concentrate the sample variability. Starting with a sample $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_K$, of size K, collected when the process is stable, the first sample principal component is the linear combination with values, $y_j = \boldsymbol{a}'(\boldsymbol{x}_j - \overline{\boldsymbol{x}})$ that has maximum sample variance among all choices with $\boldsymbol{a}'\boldsymbol{a} = 1$. The second sample principal component is the linear combination, having values $\boldsymbol{b}'(\boldsymbol{x}_j - \overline{\boldsymbol{x}})$, that has maximum variance among all linear combinations with $\boldsymbol{b}'\boldsymbol{b} = 1$ and that have zero correlation with the first principal component. The third sample principal component is the linear combination with maximum sample variance, subject to being uncorrelated with each of the first two principal components and having coefficient vector of length one. See *Johnson* and *Wichern* [18.14] for a thorough description of principal components.

The coefficients that provide the maximum variance are the eigenvectors $\boldsymbol{e}$ of the sample covariance matrix $\mathbf{S}$. That is, they are the solutions of

$$\mathbf{S}\,\boldsymbol{e} = \lambda\,\boldsymbol{e}$$

with the $\boldsymbol{e}$ normalized so that $\boldsymbol{e}'\boldsymbol{e} = 1$. There are p solutions $(\lambda_i, \boldsymbol{e}_i)$ where $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_p$. The first sample principal component

$$\hat{y}_{1j} = \boldsymbol{e}_1'(\boldsymbol{x}_j - \overline{\boldsymbol{x}})$$

has the maximum possible variance λ_1. More generally, the k-th principal component

$$\hat{y}_{kj} = \boldsymbol{e}_k'(\boldsymbol{x}_j - \overline{\boldsymbol{x}}), \quad k = 1, 2, \cdots, p$$

has sample variance λ_k.

If the process is stable over time, then the values of the first two principal components should be stable. Conversely, if the principal components remain stable over time, the common effects which influence the process are likely to remain constant.

Through an example, we introduce a two-part monitoring procedure based on principal components.

18.5.1 An Application Using Principal Components

Today, with electronic and other automated methods of data collection, major chemical and drug companies regularly measure over 100 different process variables such as temperature, pressure, concentrations and weights at various positions along the production process. Even with 11 variables to monitor, there are 55 possible pairs of variables for which ellipse format charts could be created. Consequently, we need to consider an alternative approach that both produces visual displays of important quantities and still has the sensitivity to detect special causes. Here we introduce a two-part multivariate quality control procedure that is widely applied when a large number of variables are being monitored [18.19].

The first part is an ellipse format chart to monitor the first two principal components. The second part is a T^2 chart based on the remaining principal components.

If the p variables have quite different variances, it is usual to standardize the variables before finding the principal components. This is equivalent to extracting the eigenvectors from the correlation matrix $\mathbf{R}$. Here we skip that step because the variables are comparable.

The values of the first and second sample principal components for the $n-$th observation are

$$\hat{y}_{1n} = \boldsymbol{e}_1'(\boldsymbol{x}_n - \overline{\boldsymbol{x}}) ,$$
$$\hat{y}_{2n} = \boldsymbol{e}_2'(\boldsymbol{x}_n - \overline{\boldsymbol{x}}) .$$

The first part of the multivariate quality control procedure is to construct a ellipse format chart for the pairs of values $(\hat{y}_{1n}, \hat{y}_{2n})$, $n = 1, 2, \ldots$.

Recall that the variance of the first principal component $\hat{y}_1$ is λ_1, that of the second principal component $\hat{y}_2$ is λ_2, and the two are uncorrelated. Consequently, the control format ellipse reduces to the collection of possible values $(\hat{y}_1, \hat{y}_2)$ such that

$$\frac{\hat{y}_1^2}{\lambda_1} + \frac{\hat{y}_2^2}{\lambda_2} \le \chi_2^2(\alpha) .$$

Chart 1: The elliptical control region for the first two principal components

Refer to the automotive assembly data on $p = 6$ variables. We base our estimate $\mathbf{S}$ of $\boldsymbol{\Sigma}$ on the first 30 stable observations. A computer calculation gives the eigenvalues and eigenvectors in Table 18.3.

The 99% ellipse format chart for the first two principal components

$$\frac{\hat{y}_1^2}{\lambda_1} + \frac{\hat{y}_2^2}{\lambda_2} \le \chi_2^2(0.01)$$

is shown in Fig. 18.16 along with the pairs of values of the principal components for the first 30 observations as well as the additional cases. There are no points out of control.

Special causes may still produce shocks to the system not apparent in the values of the first two principal components and a second chart is required.

Chart 2: A T^2 chart for the remaining principal components

For the 30 stable observations, the approximation to $\boldsymbol{x}_j - \overline{\boldsymbol{x}}$ by the first two principal components has the form $\hat{y}_{1j}\boldsymbol{e}_1 + \hat{y}_{1j}\boldsymbol{e}_2$ [18.14]. This leaves an unexplained component of the j-th deviation $\boldsymbol{x}_j - \overline{\boldsymbol{x}}$. Namely,

$$\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2 .$$

For each j, this unexplained component is perpendicular to both of the eigenvectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Consequently,

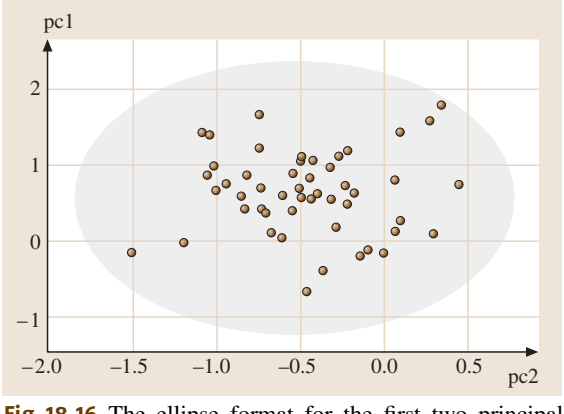


Fig. 18.16 The ellipse format for the first two principal components – automotive data

Table 18.3 Eigenvectors and eigenvalues from the 30 stable observations

	e_1	e_2	e_3	e_4	e_5	e_6
x_1	0.1193	0.4691	0.0752	0.2906	0.2672	0.7773
x_2	0.1295	0.4576	0.2508	0.6237	0.0366	−0.5661
x_3	0.1432	0.7170	−0.1161	−0.6323	−0.1746	−0.1471
x_4	0.0964	0.0529	−0.9555	0.2568	0.0619	−0.0715
x_5	0.9677	−0.2312	0.0700	−0.0641	0.0322	−0.0030
x_6	0.0517	0.0135	−0.0078	0.2380	−0.9444	0.2204
λ_i	0.3544	0.1864	0.1076	0.0972	0.0333	0.0088

it is perpendicular to its approximation $\hat{y}_{1j}\boldsymbol{e}_1 + \hat{y}_{1j}\boldsymbol{e}_2$, which implies that the approximation and unexplained component have zero sample covariance.

Seen another way, let $\mathbf{E} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_p)$ be the orthogonal matrix whose columns are the eigenvectors of $\mathbf{S}$. The orthogonal transformation of the unexplained part is

$$\mathbf{E}'(\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2) = \begin{pmatrix} 0 \\ 0 \\ \hat{y}_{3j} \\ \cdot \\ \cdot \\ \cdot \\ \hat{y}_{pj} \end{pmatrix} .$$

The first two components are always zero so we base the T^2 chart on the values of the last $p-2$ principal components. Because the

$$\text{sample variance } (\hat{y}_{ij}) = \lambda_i \text{ for } i = 1, 2, 3, \cdots, p$$

and the principal components have zero sample covariances, the T^2 based on the original quantities $\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2$ is equivalent to the one based on the values

$$T_j^2 = \frac{\hat{y}_{3j}^2}{\lambda_3} + \frac{\hat{y}_{4j}^2}{\lambda_4} + \cdots + \frac{\hat{y}_{pj}^2}{\lambda_p} .$$

Because the coefficients of the linear combinations, $\boldsymbol{e}_i$, are also estimates, the principal components do not have a normal distribution even when the underlying population is normal. Consequently, it is customary to use the large-sample approximation to the upper control limit, UCL $= \chi^2_{p-2}(\alpha)$.

This T^2 statistic can be based on high-dimensional data. When $p = 20$ variables are measured, it concerns the 18-dimensional space perpendicular to the first two eigenvectors. Still, it is reported as highly effective in picking up special causes.

Refer the automobile assembly data. The quality ellipse for the first two principal components was shown in Fig. 18.16. To illustrate the second step of the two step monitoring procedure, we create the chart for the other variables. Since $p = 6$, this 99% chart is based on $6-2 = 4$ dimensions and the upper control limit is $\chi^2_4(0.01) = 13.28$. We plot the time sequence of values

$$T_n^2 = \frac{\hat{y}_{3n}^2}{\lambda_3} + \frac{\hat{y}_{4n}^2}{\lambda_4} + \frac{\hat{y}_{5n}^2}{\lambda_5} + \frac{\hat{y}_{6n}^2}{\lambda_6} .$$

The T^2 chart is shown as Fig. 18.17.

Something has likely happened at the 33rd, 36th, 37th and the 39th observations. For the 39th observation, the values of the last principal components are -0.2712, -0.2105, 0.8663, -0.2745, respectively. The value of $\hat{y}_{5,39} = 0.8663$ is particularly large, with reference to the coefficient vector $\boldsymbol{e}_5$ in Table 18.3, we see that the fifth principal component is essentially X_6. From the data and the univariate Shewhart X-bar charts in Fig. 18.11, we see that the mean of the last variable has increased.

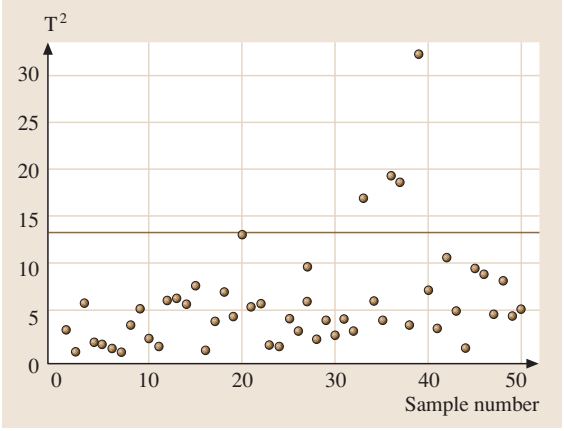


Fig. 18.17 T^2 chart based on the last four principal components – automotive data

In some applications in the pharmaceutical industry hundreds of variables are monitored. Then, the space orthogonal to the first few principal components has dimension greater than 100 and some the eigenvalues are very small. An alternative approach [18.19], which avoids the difficulty of dividing by very small numbers, has been successful applied. For each of the K stable observations, take the sum of squares of its unexplained component

$$d^2_{\perp j} = (\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2)' \times (\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2) \,.$$

Note that, by inserting $\mathbf{E}\mathbf{E}' = \mathbf{I}$, we also have

$$d^2_{\perp j} = \left(\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2\right)' \mathbf{E}\mathbf{E}' \times \left(\boldsymbol{x}_j - \overline{\boldsymbol{x}} - \hat{y}_{1j}\boldsymbol{e}_1 - \hat{y}_{2j}\boldsymbol{e}_2\right) = \sum_{k=3}^{p} \hat{y}^2_{kj}$$

which is just the sum of squares of the neglected principal components.

Using either form, the $d^2_{\perp n}$ are plotted versus n to create a control chart. The lower control limit is zero and the upper limit is set by approximating the distribution of $d^2_{\perp n}$ as a constant c times a chi-square random variable with ν degrees of freedom. The constant c and degrees of freedom ν are chosen to match the sample mean and variance of the $d^2_{\perp j}$, $j = 1, \cdots, K$. In particular, setting

$$\overline{d^2_{\perp}} = \frac{1}{K}\sum_{j=1}^{K} d^2_{\perp j} = c\,\nu \,,$$

$$s_{dd} = \frac{1}{K}\sum_{j=1}^{K}\left(d^2_{\perp j} - \overline{d^2_{\perp}}\right)^2 = 2c^2\nu \,,$$

we determine that

$$c = \frac{s_{dd}}{2\overline{d^2_{\perp}}} \quad \text{and} \quad \nu = 2\frac{(\overline{d^2_{\perp}})^2}{s_{dd}} \,.$$

The upper control limit is then $c\chi^2_\nu(\alpha)$, where $\alpha = 0.05$ or 0.01.

We remark that this two-step procedure can be made more sensitive by using, for instance, Crosier's scheme.

18.6 Difficulties of Time Dependence in the Sequence of Observations

We must include a warning that pertains to the application of any of the quality monitoring procedures we have discussed. They are all based on the assumption that the observations are independent. Most often, especially with automated collection procedures, observations may be taken close together in time or space. This can produce a series of observations that are not independent but correlated in time.

We emphasize that the methods described in this chapter are based on the assumption that the multivariate observations $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$ are independent of one another. The presence of even a moderate amount of time dependence among the observations can cause serious difficulties for monitoring procedures.

One common and simple univariate model that usually captures much of the time dependence is a first-order autoregressive process (AR(1))

$$X_n - \mu = \phi\,(\,X_{n-1} - \mu\,) + \varepsilon_n \,,$$

where $-1 < \phi < 1$. The ε_n are independent, mean-zero, shocks all having the same variance σ^2_ε. The AR(1) model relates the observation at time n, to the observation at time $n-1$, through the coefficient ϕ. The name autoregressive model comes from the fact that the model looks like a regression model with X_n as the dependent variable and the previous value X_{n-1} as the independent variable.

Under normality of errors, if $\phi = 0$, the autoregressive model implies that the observations are independent. It is not at all unusual in practice to find values of ϕ as high as 0.3 or 0.4.

The AR(1) model, implies that all of the X_n have the same variance $\sigma^2_X = \sigma^2_\varepsilon / (1-\phi^2)$. As shown in *Johnson* and *Langeland* [18.20], if the sample variance s^2 is calculated from a long series of adjacent observations, s^2 will closely approximate σ^2_X. That is, the correct variance is being estimated.

We first consider the effects of dependence in the context of the X-bar chart. When the individual observations X_n are plotted, with limits set at $\pm 3\sigma_X$, we still have

$$P(\,|\,X_n - \mu\,|\,> 3\sigma_X\,) = 0.0027 \,.$$

The observations are correlated but we expect the same number of false alarms.

In this sense, the X-bar chart is robust with respect to dependence. If the points being plotted, $\overline{X}_n$, correspond

Table 18.4 Probability of false alarms when the process is in control. Normal populations and X-bar chart

	$P\left(\lvert\overline{X}_n - \mu\rvert > 3\sigma_X/\sqrt{m}\right)$				
	$\phi = -0.6$	$\phi = -0.3$	$\phi = 0$	$\phi = 0.3$	$\phi = 0.6$
$m = 1$	0.0027	0.0027	0.0027	0.0027	0.0027
$m = 3$	0.0000	0.0003	0.0027	0.0147	0.0484
$m = 5$	0.0000	0.0002	0.0027	0.0217	0.0971

Table 18.5 The estimated ARL for Page's CUSUM when the process is in control. Normal populations

	$\phi = -0.4$	$\phi = -0.2$	$\phi = -0.1$	$\phi = 0$	$\phi = 0.1$	$\phi = 0.2$	$\phi = 0.4$
$k = 0.5,\ \ h = 4.24$	9549.0	884.0	390.5	206.1	121.8	82.6	43.9
$k = 0.75,\ \ h = 2.96$	2901.2	706.3	356.2	204.1	128.5	88.5	48.1

Table 18.6 The h value to get in-control ARL ≈ 200, $k = 0.5$. Page's CUSUM

	$\phi = -0.6$	$\phi = -0.3$	$\phi = -0.1$	$\phi = 0$	$\phi = 0.1$	$\phi = 0.3$	$\phi = 0.6$
h	2.34	2.94	3.77	4.24	4.76	6.38	10.6
in-control ARL	214.6	205.5	213.2	206.1	202.7	205.8	210.3

to the sample mean of m adjacent observations, then

$$\sigma^2_{\overline{X}} = \mathrm{Var}(\overline{X}_n) = \frac{\sigma^2_X}{m}\left(1 + \sum_{j=1}^{m-1}\frac{m-j}{m}\phi^j\right).$$

In this case, the probability of a false alarm when the limits are set at $3\sigma_X/\sqrt{m}$, is

$$P\left(\lvert\overline{X}_n - \mu\rvert > 3\frac{\sigma_X}{\sqrt{m}}\right) = P\left(\left|\frac{\overline{X}_n - \mu}{\sigma_{\overline{X}}}\right| > 3\frac{\sigma_X/\sqrt{m}}{\sigma_{\overline{X}}}\right).$$

There are some dramatic changes from the nominal value of the probability of a false alarm when the process is in control. Some values are given in Table 18.4.

The consequences of dependence are much more severe on the CUSUM statistic. *Johnson* and *Bagshaw* [18.21] present some limiting results for the distribution of ARL when the centering values are selected so the contribution of the n-th observation has mean zero for every n. The presence of dependence greatly alters the ARL. Essentially, this occurs because the CUSUM is a smooth function of the stochastic process defined at $n = 0, 1, \cdots$ by

$$n^{-1/2}S_n = n^{-1/2}\sum_{i=1}^{n}(X_i - \mu).$$

For large fixed n, $n^{-1/2}S_n$ is normal with variance approaching $\sigma^2_X(1+\phi)/(1-\phi)$. This is quite different from σ^2_X, which is the value if we ignore the dependence in the model.

Table 18.5 gives the estimated ARL using Page's CUSUM procedure, for in-control normal populations, for a few values of ϕ. The h values in the CUSUM scheme are chosen carefully so that the in-control ARL at $\phi = 0$ is around 200. The ARL values are based on 5000 trials and apply to the situation where the dependence is not noticed when $\phi \neq 0$, but that variance is estimated by the usual formula using a long series of consecutive observations.

We see dramatic reductions in ARL even for small positive values of ϕ. To be able to get the desired in-control ARL when there is time dependence in the sequence of observations, the h values in the control schemes have to be changed. Table 18.6 presents the h values obtained from simulation to get in-control ARL near 200, for different values of ϕ, using Page's CUSUM scheme when the dependence is not noticed.

In the context of multivariate procedures, dependence can often be represented as a multivariate first-order autoregressive model. Let the $p \times 1$ random vector $\boldsymbol{X}_t$ follow the multivariate AR(1) model

$$\boldsymbol{X}_n - \boldsymbol{\mu} = \boldsymbol{\Phi}(\boldsymbol{X}_{n-1} - \boldsymbol{\mu}) + \varepsilon_n \qquad (18.9)$$

where the ε_n are independent and identically distributed with $E(\varepsilon_n) = \boldsymbol{0}$ and $\mathrm{Cov}(\varepsilon_n) = \boldsymbol{\Sigma}_\varepsilon$ and all of the eigenvalues of $\boldsymbol{\Phi}$ are between -1 and 1. Under this model

$$\mathrm{Cov}(\boldsymbol{X}_n, \boldsymbol{X}_{n-j}) = \boldsymbol{\Phi}^j\boldsymbol{\Sigma}_X,$$

$$\text{where } \boldsymbol{\Sigma}_X = \sum_{j=0}^{\infty}\boldsymbol{\Phi}^j\boldsymbol{\Sigma}_\varepsilon\boldsymbol{\Phi}'^j.$$

Table 18.7 The estimate in-control ARL using Crosier's multivariate scheme

	$\phi = -0.3$	$\phi = -0.2$	$\phi = -0.1$	$\phi = 0$	$\phi = 0.1$	$\phi = 0.2$	$\phi = 0.3$
$k = 0.5,\ h = 5.55$	4881.3	1295.2	454.8	203.6	105.5	65.6	43.2

The multivariate AR(1) model relates the observation at time n to the observation at time $n-1$ through the coefficient matrix Φ. Further, the autoregressive model says the observations are independent, under multivariate normality, if all the entries in the coefficient matrix Φ are 0.

As shown in *Johnson* and *Langeland* [18.20],

$$\overline{\boldsymbol{X}} \to^{P} \boldsymbol{\mu}\ ,$$

$$\boldsymbol{S} = \frac{1}{n-1}\sum_{t=1}^{n}(\boldsymbol{X}_t - \overline{\boldsymbol{X}})(\boldsymbol{X}_t - \overline{\boldsymbol{X}})' \to^{P} \boldsymbol{\Sigma}_X\ ,$$

where the arrow indicates convergence in probability.

Similar to the X-bar chart, the T^2 chart is robust with respect to dependence when individual observations are plotted. The probability of a false alarm at any time n is still

$$P\left[(\boldsymbol{X}_n - \boldsymbol{\mu})'\boldsymbol{\Sigma}_X^{-1}(\boldsymbol{X}_n - \boldsymbol{\mu}) \geq \chi_p^2(0.01)\right] = 0.01$$

since, under normality, $\boldsymbol{X}_n - \boldsymbol{\mu}$ has a $p-$variate normal distribution with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{\Sigma}_X$. If the observations are independent, and the process is in control, the probability is 0.01 that a single observation will falsely signal that a change has occurred. The ARL is then $1/0.01 = 100$.

When the plotted points correspond to the sample mean of m adjacent observations, the situation is more complicated. To simplify our calculations, we consider the case where has $\boldsymbol{\Phi} = \phi\mathbf{I}$, where $|\phi| < 1$. Consider the multivariate T^2 chart where the process is considered to be in control if

$$m(\overline{\boldsymbol{X}}_n - \boldsymbol{\mu})'\Sigma_X^{-1}(\overline{\boldsymbol{X}}_n - \boldsymbol{\mu}) \leq \chi_p^2(0.01)\ .$$

If the observations are independent, and the process is in control, the probability is 0.01 that a single observation will falsely signal that a change has occurred. The ARL is $1/0.01 = 100$. Using $\chi_p^2(0.005)$ as limit gives an ARL of $1/0.005 = 200$ when the process is in control. However, if the observations are related by our simplified multivariate AR(1) model, the average of m adjacent observations has covariance matrix

$$\mathrm{Cov}(\overline{\boldsymbol{X}}_n) = \frac{1}{m}\boldsymbol{\Sigma}_X\left(1 + \sum_{j=1}^{m-1}\frac{m-j}{m}\phi^j\right)\ .$$

not $\boldsymbol{\Sigma}_X/m$, and this will cause some change from the nominal probability.

The multivariate CUSUM statistics are all based on the stochastic process, defined at $n = 1, 2, \cdots$ by

$$n^{-1/2}\boldsymbol{S}_n = n^{-1/2}\sum_{j=1}^{n}(\boldsymbol{X}_j - \boldsymbol{\mu})$$

whose covariance at time n

$$\mathrm{Cov}\left(n^{-1/2}\sum_{t=1}^{n}\boldsymbol{X}_t\right) \to (\boldsymbol{I} - \Phi)^{-1}\boldsymbol{\Sigma}_X + \boldsymbol{\Sigma}_X(\boldsymbol{I} - \Phi')^{-1} - \boldsymbol{\Sigma}_X\ .$$

This can be considerably different from the covariance $\boldsymbol{\Sigma}_X$ used when dependence is ignored.

Table 18.7 provides the estimated ARL for in-control normal populations with covariance matrix $\boldsymbol{\Sigma} = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix}$, using Crosier's CUSUM scheme. The ARL values are based on 5000 trials where the covariance matrix is estimated by $\boldsymbol{S}$ using a long series of consecutive observations.

For more details on large-sample approximations, including a limit for Crosiers statistic, see *Li* [18.6].

Based on the calculations above and consideration of other cases, we must emphasize that the independence assumption is crucial to all of the procedures we discussed that are based on cumulative sums. The results based on this assumption can be seriously misleading if the observations are, in fact, even moderately dependent.

The best approach, when dependence is identified as being present, is to fit a time-series model. Then, as suggested in *Bagshaw* and *Johnson* [18.22], the residuals can be monitored for a shift. In the univariate case a CUSUM statistic can be applied. See also *Hawkins* and *Olwell* [18.23], Section 9.3.

References

18.1 W. A. Shewhart: *Economic Control of Quality of Manufactured Product* (Van Nostrand, New York 1931)

18.2 E. S. Page: Continuous inspection schemes, Biometrika **41**, 100–115 (1954)

18.3 R. B. Crosier: A new two-sided cumulative sum quality control scheme, Technometrics **28**, 187–194 (1986)

18.4 J. M. Lucas, M. S. Saccucci: Exponentially weighted moving average control schemes: properties and enhancements, Technometrics **32**, 1–12 (1990)

18.5 D. C. Montgomery: *Introduction to Statistical Quality Control*, 4th edn. (Wiley, New York 2000)

18.6 Li, R. New Multivariate Schemes for Statistical Process Control, Dissertation, Department of Statistics, Univ. of Wisconsin (2004)

18.7 J. E. Jackson: Quality control methods for several related variables, Technometrics **1**, 359–377 (1959)

18.8 J. E. Jackson: Multivariate quality control, Commun. Stat. A **14**, 2657–2688 (1985)

18.9 N. Doganaksoy, J. Fulton, W. T. Tucker: Identification of out of control quality characteristics in multivariate manufacturing environment, Commun. Stat. A **20**, 2775–2790 (1991)

18.10 N. D. Tracy, J. C. Young, R. L. Mason: Multivariate quality control charts for individual observations, J. Qual. Technol. **24**, 88–95 (1992)

18.11 R. B. Crosier: Multivariate generalizations of cumulative sum quality-control schemes, Technometrics **30**, 291–303 (1988)

18.12 C. Fuchs, R. S. Kenett: *Multivariate Quality Control: Theory and Applications* (Marcel Dekker, New York 1998)

18.13 K. Yang, J. Trewn: *Multivariate Statistical Methods in Quality Management* (McGraw-Hill, New York 2004)

18.14 R. A. Johnson, D. W. Wichern: *Applied Multivariate Statistical Analysis* (Prentice Hall, Piscataway 2002)

18.15 C. A. Lowry, W. H. Woodall, C. W. Champ, S. E. Rigdon: A multivariate exponentially weighted moving average control chart, Technometrics **34**, 46–53 (1992)

18.16 D. Ceglarek, J. Shi: Dimensional variation reduction for automotive body assembly, Manuf. Rev. **8**, 139–154 (1995)

18.17 J. J. Pignatiello, G. C. Runger: Comparisons of multivariate CUSUM charts, J. Qual. Technol. **22**, 173–186 (1990)

18.18 C. A. Lowry, D. C. Montgomery: A review of multivariate control charts, IIE Trans. **27**, 800–810 (1995)

18.19 T. Kourti, J. F. MacGregor: Multivariate SPC methods for process and product monitoring, J. Qual. Technol. **28**, 409–428 (1996)

18.20 R. A. Johnson, T. Langeland: A linear combinations test for detecting serial correlation in multivariate samples. In: *Statistical Dependence*, Topics, ed. by H. Block et al. (Inst. Math. Stat. Mon. 1991) pp. 299–313

18.21 R. A. Johnson, M. Bagshaw: The effect of serial correlation on the performance of CUSUM tests, Technometrics **16**, 103–112 (1974)

18.22 M. Bagshaw, R. A. Johnson: Sequential procedures for detecting parameter changes in a time-series model, J. Am. Stat. Assoc. **72**, 593–597 (1977)

18.23 D. M. Hawkins, D. Olwell: *Cumulative Sum Charts and Charting for Quality Improvement* (Springer, New York 1998)

Reliability Part C

Part C Reliability Models and Survival Analysis

Part C focuses on reliability models, statistical accelerated testing and survival analysis. The first five chapters in this part emphasize general system modeling while the last four chapters emphasize software systems. The first chapter in this part, Chapt. 19, discusses variations of the statistical survival model and the step-stress accelerated-failure-time model and their important applications to both biomedical and engineering studies, followed by Chapt. 20, which focuses on failure-rate modeling with respect to heterogeneous populations and presents the concepts and properties of mixture failure rates, proportional hazards, additive hazards and accelerated life tests in heterogeneous populations. Chapter 21 provides an overview of various proportional hazard regression (PHR) models, including the stratified Cox model, the Cox model with time-dependent covariates and various hypothesis-testing methods for validating PHR models. Several extended models such as nonproportional random effects are also discussed. Chapter 22 outlines various statistical models for describing lifetime distributions, such as the log-normal and Weibull, with the inclusion of multiple accelerating variables in accelerated-life tests, and discusses some of the potential difficulties of accelerated testing in practice. Chapter 23 details the statistical methods for designing various types of accelerated reliability tests, including constant-stress tests, step-stress tests, and step-stress accelerated degradation tests under harsher environments with multiple-step stress levels.

The next four chapters focus on statistical models in software systems, starting with Chapt. 24, which focuses on aspects of technology evolution for high-assurance systems, including dynamic verification and validation, reliability and security issues, safety assurance, automated dependency analysis, model checking on system specifications and model checking based on test-case generation. Chapter 25 discusses in detail and reviews software reliability growth modeling and optimization problems including software testing effort, growth models, parameter estimation, optimal release policies, and resource allocation, while Chapt. 26 focuses on a software development design-review process based on a quality engineering approach to analyze the relationships among the quality of the design-review activities, including software reliability and human factors in the development process. Chapter 27 discusses a generalized prediction model based on a nonhomogeneous Poisson process framework for evaluating the reliability and its corresponding confidence intervals for software systems in a random field environment.

19. Statistical Survival Analysis with Applications

This chapter discusses several important and interesting applications of statistical survival analysis which are relevant to both medical studies and reliability studies. Although it seems to be true that the proportional hazards models have been more extensively used in the application of biomedical research, the accelerated failure time models are much more popular in engineering and reliability research. Through several applications, this chapter not only offers some unified approaches to statistical survival analysis in biomedical research and reliability/engineering studies, but also sets up necessary connections between the statistical survival models used by biostatisticians and those used by statisticians working in engineering and reliability studies. The first application is the determination of sample size in a typical clinical trial when the mean or a certain percentile of the survival distribution is to be compared. The approach to the problem is based on an accelerated failure time model and therefore can have direct application in designing reliability studies to compare the reliability of two or more groups of differentially manufactured items. The other application we discuss in this chapter is the statistical analysis of reliability data collected from several variations of step-stress accelerated life test. The approach to the problem is based on the accelerated failure time model, but we will point out that these methodologies can be directly applied to medical and clinical studies when different doses of a therapeutic compound are administered in a sequential order to experimental subjects.

Failure time data or survival data are frequently encountered in biomedical studies, engineering, and reliability research. Applications of lifetime distribution methodologies range from investigations into the reliability of manufactured items to research involving human diseases. In medical studies, clinical endpoints for assessment of efficacy and safety of a promising therapy usually include occurrence of some predefined events such as deaths, the onset of a specific disease, the response to a new chemotherapy in treatment of some advanced cancer, the eradication of an infection caused by a certain microorganism, or serious adverse events. In engineering and reliability studies, manufactured items such as mechanical or electronic components are often subjected to life tests to obtain information on their endurance. This involves putting items into operation, often in a laboratory setting, and observing them until they fail. For example, *Nelson* [19.1] described a life test experiment in which specimens of a type of electronic insulating fluid were subjected to a constant voltage stress. The length of time until each specimen broke down was observed and investigated in its association with the voltage level. In all of the studies mentioned above, the primary variable of interest is usually the *survival time* to the occurrence of a specific predetermined event. One of the important features in survival data encountered

in both medical research and engineering studies is the existence of censored observations when only a lower (or upper) bound of the failure time on some experimental units are available. Censoring occurs frequently because of time limits and other restrictions during the process of data collection. In a life test experiment of manufactured items, for example, it may not be feasible to continue experimentation until all items under study have failed. If the study is terminated before all have failed, then for items that have not failed at the time of termination only a lower bound on lifetime is available.

The statistical analysis of survival data has been well developed in the literature. The estimation of the survival distribution can be done by the Kaplan–Meier product-limit estimator [19.2], which can also be viewed as a kind of nonparametric maximum likelihood estimator [19.3]. For studies in which the aim is to compare the survival distribution of two groups of subjects, the logrank test has been the most common method, although other rank tests such as the generalized Wilcoxon test are also used [19.2]. The logrank test can also be extended to allow an adjustment to be made for other covariates [19.4]. The major developments in the analysis of survival data have focused on several families of survival distributions. Two very important models of survival distribution are the model of proportional hazards and the accelerated failure time model. The proportional hazard model is a regression method introduced by *Cox* [19.5], which can be used to investigate the effects of several covariates on survival distribution at the same time. Cox's method is a semi-parametric approach – no particular type of distribution is assumed for the survival data, but a strong assumption is made on the effects of differences, which is referred to as the assumption of proportional hazards. Regression diagnostic procedures are also available to assess the assumption of proportional hazards [19.6,7], and some tests of the assumption of proportional hazards are also introduced through the incorporation of time-dependent covariates [19.8]. Extensions to Cox's proportional hazards model are the analysis of residuals, time-dependent coefficient, multiple/correlated observations, time-dependent strata, and estimation of underlying hazard function [19.8–10]. The accelerated failure time model, on the other hand, assumes that the covariates act by expanding or contracting time by a factor which is the exponential of a linear function of the covariates. In the logarithmic scale of the survival time, the accelerated failure time model is essentially a scale-location family of distributions.

It is quite interesting to observe that the proportional hazards models have been more extensively used in the application of biomedical research, while the accelerated failure time models are much more popular in engineering and reliability research. Part of the reason that Cox's proportional hazards models are popular in biomedical studies is the very fact that the assumption of proportional hazards summarizes the risk factor for a specific disease into a single quantity, the hazard ratio, which makes the interpretation easy to understand for clinicians. As an example, medical literature has demonstrated that a key protein, apolipoprotein E4 (ApoE4), contributes to the development of Alzheimer's disease [19.11]. Clinicians are interested in knowing how much the risk of Alzheimer's disease is increased for ApoE4-positive subjects compared to ApoE4-negative subjects. The point and confidence interval estimate to the hazard ratio associated with ApoE4 will adequately address the question if the assumption of proportional hazards can be adequately verified. On the other hand, the accelerated failure time models often make very good sense when the multiplicative time scale is assumed based on the level of covariate. As an example, assume that the lifetime of photocopiers has a hazard function that is a function of the total number of copies made, but the data on their failures were recorded in calendar time. Covariates that were related to the number of copies per day might be very successful in an accelerated failure time model. If the underlying hazard function had a particular form, say a sharp upturn after 25 000 cycles, a proportional hazard model would not fit as well. Similar examples can also be found for biological data related to cumulative toxicity or other damage.

Whether or not a statistical model is appropriate in a specific application depends on the distributional property of the observed variable and the specific research questions to be addressed. This chapter focuses on several applications of survival analysis in both medical/biological research and engineering/reliability research. We discuss several interesting applications which are relevant to both medical studies and reliability studies. The first application is the determination of sample size in a typical clinical trial when the mean or a certain percentile of the survival distribution is to be compared. The approach to the problem is based on an accelerated failure time model and therefore can have direct application in designing reliability studies to compare the reliability of two or more groups of differentially manufactured items. The other application we discuss in this chapter is the statistical

analysis of reliability data collected from several variations of step-stress accelerated life test. The approach to the problem is based on the accelerated failure time model, but we will point out that these methodologies can be directly applied to medical and clinical studies when different doses of a therapeutic compound are administered in a sequential order to experimental subjects.

19.1 Sample Size Determination to Compare Mean or Percentile of Two Lifetime Distributions

The determination of sample size is an important subject in planning long-term medical trials. In a randomized longitudinal clinical trial involving a treatment group and a control group, if the survival time to a particular event (e.g., death, relapse of symptoms) is the primary concern for the study, there are two important types of comparisons between the treatment group and the control group. One is the comparison of two survival curves and the other is the comparison of two common survival characteristics such as two means and two percentiles. Although the comparison of two survival curves is the major interest in many studies, the comparison of two means or two percentiles is important in many other applications. For example, in the announcement of the aging intervention testing program (RFA-AG-02-005), the National Institute on Aging (NIA) of USA states that one of the major research objectives of this program is to identify interventions that increase mean life expectancy by 10% in phase I studies, which may be terminated at 50% survivorship. This type of aging intervention study based on animal models has recently received much attention in the community of aging research. For example, caloric restriction has been identified as an intervention that extends the life span of both mammalian animal models and a variety of invertebrate animal models [19.12]. Mutations in the *dw* and *df* genes have been shown to attenuate the rate of aging in mice [19.13, 14]. *Warner* et al. [19.15] provided more details of biological interventions to promote healthy aging. The sample size computation for this type of study requires a statistical test that compares the mean lifetime between the control group and the treatment group based on type II censored observations.

Sample size determination methods are always based on certain parametric or semiparametric statistical models. This section concerns two important families of distributions used in the analysis of lifetime data: one is the family of proportional hazards and the other is the location-scale family of the log-transformed lifetime. The traditional approach to the sample size problem in planning long-term medical trials is based on the logrank test for the comparison of two lifetime distributions between the control group and the treatment group. Although the logrank test can be derived from both the proportional hazards family and the location-scale family of log-transformed lifetime, it is the proportional hazards family that most sample size computation methods with logrank test in the literature have been based on. In fact, the statistics literature on sample size calculation for failure time data is almost entirely devoted to tests based on exponential survival distributions [19.16–18] or binomial distributions [19.19, 20]. This is largely due to the fact that with the more general conditions hazard functions and ratios are no longer constant over time, so that the usual tests based on exponential models with constant hazard ratios no longer apply. *Schoenfeld* [19.21] and *Freedman* [19.22] presented methods for sample size calculation based on the asymptotic expectation and variance of the logrank statistic and the assumption of proportional hazards. *Lakatos* [19.23] proposed a Markov model to estimate the sample sizes for the comparison of lifetime distributions based on the logrank test. *Wu* et al. [19.24] provided a sample size computation method that allows time-dependent event (dropout) rate. *Lakatos* and *Lan* [19.25] compared several sample size calculation methods associated with the logrank test.

When the primary concern in a medical or reliability study is to compare the means or certain percentiles of two lifetime distributions, such as in the aging intervention testing program, the sample size determination based on proportional hazards assumption runs into the problem of expressing the difference or ratio of two means or two percentiles of lifetime distributions into the ratio of two hazard functions. Although this is no problem with exponential distributions or Weibull distributions with the same shape parameter, it might not always be possible for other families of proportional hazards. On the other hand, the location-scale family of log-transformed lifetime seems to be a very natural family of lifetime distributions to use for this type of sample size problem. This is based on the fact that,

when the scale parameters are assumed to be the same across different groups, the comparison of the means in lifetimes is equivalent to the comparison of location parameters. Although nonparametric tests such as the logrank test are appealing when the underlying lifetime distributions are unknown, they tend to be less efficient when the pilot information suggests a certain family of lifetime distributions and such information is ignored based on nonparametric tests. In fact, the logrank test bears 100% asymptotic relative efficiency when there is no censoring or when there is random but equal censoring in two groups within the family of Weibull distributions [19.26]. When data are from lognormal distributions differing only with respect to location, however, the asymptotic relative efficiency of the logrank test decreases to 82% [19.3].

Another common feature of most sample size determination methods in the literature is that they all deal with type I censored samples in which a prespecified time is used to terminate the experiment. This section studies the sample size determination to compare the means or certain percentiles of two lifetime distributions when both samples are subject to type II censoring. Our approach is based on a location-scale family of log-transformed lifetime distributions and the asymptotic normality of maximum likelihood estimates (MLEs). We also apply our methods to both the family of lognormal distributions and the family of Weibull distributions and compare our method with other well-known methods such as those based on *Rubinstein* et al. [19.18] and *Freedman* [19.22].

Although we will discuss the sample size determination in the context of designing a medical and biological study in this section, the basic ideas and results can be readily applied to the design of various engineering studies to compare the reliability of different groups of manufactured items. In fact, the application of our proposed methods to designing engineering studies will be even more intuitive to engineers as our approach is based on a location-scale family of log-transformed lifetime distributions, which is essentially equivalent to the accelerated failure time model popularly used in engineering and reliability studies.

19.1.1 The Model and Sample Size

Let T_c be the survival time for the control group, and T_t be the survival time for the treatment group. Assume that $Y_i = \ln T_i$ follows a probability distribution that belongs to a location-scale family with probability density function $\frac{1}{\sigma_i} g\left(\frac{y-\mu_i}{\sigma_i}\right)$, $-\infty < y < \infty$, $i = c, t$, where $g(s) > 0$ is a differentiable positive function whose derivative $g'(s) = \frac{dg(s)}{ds}$ makes all integrations used in this chapter exist. Let $G(s) = \int_{-\infty}^{s} g(t)\,dt$ be the cumulative distribution of $g(s)$. We also assume that both σ_c and σ_t are given and that $\sigma_c = \sigma_t = \sigma > 0$. The mean of $T_i = \exp(Y_i)$ is

$$ET_i = \int_{-\infty}^{\infty} e^y \frac{1}{\sigma} g\left(\frac{y-\mu_i}{\sigma}\right) dy = e^{\mu_i} \int_{-\infty}^{\infty} e^{\sigma s} g(s)\,ds\,, \tag{19.1}$$

for $i = c, t$. Hence,

$$\frac{ET_t}{ET_c} = e^{\mu_t - \mu_c}\,. \tag{19.2}$$

Let $0 < \delta < 1$. For $i = c, t$, a straightforward integration gives the $100\delta\%$ percentile of T_i as

$$\tau_i(\delta) = e^{\mu_i + \sigma G^{-1}(\delta)}\,, \tag{19.3}$$

where G^{-1} is the inverse function of G. It follows that

$$\frac{\tau_t(\delta)}{\tau_c(\delta)} = e^{\mu_t - \mu_c}\,. \tag{19.4}$$

Therefore, the problem of testing the ratio of two means or two percentiles between the control group and the treatment group can always be reduced to the problem of testing the difference between μ_t and μ_c.

Suppose that two independent samples of size n_c and n_t are drawn from the distributions of T_c and T_t, respectively. For $i = c, t$, we assume that only the smallest $100\,q_i\%$ of the samples are observed for some given $0 < q_i < 1$. If we let $r_i = [q_i n_i]$, then only the order statistics up to r_i-th are observed for group $i = c, t$. Let γ be the ratio of two sample sizes: $\gamma = \frac{n_t}{n_c}$. We want to decide the sample sizes for testing the null hypothesis $H_0 : \mu_c = \mu_t$ against the alternative $H_1 : \mu_c \neq \mu_t$ at an asymptotic significance level α $(0 < \alpha < 1)$. If this test is to achieve $100(1-\beta)\%$ power to detect a difference of $d = \mu_t - \mu_c$, the required sample size n_c for the control group is the unique solution to the following equation:

$$\beta = \Phi\left(z_{\alpha/2} - \frac{d}{\sqrt{\frac{1}{n_c}\left(\frac{1}{K_c^2} + \frac{1}{\gamma K_t^2}\right)}}\right) - \Phi\left(-z_{\alpha/2} - \frac{d}{\sqrt{\frac{1}{n_c}\left(\frac{1}{K_c^2} + \frac{1}{\gamma K_t^2}\right)}}\right)\,, \tag{19.5}$$

where $z_{\alpha/2}$ is the upper $\alpha/2$ percentage point of the standard normal distribution, Φ is the cumulative distribution function of the standard normal distribution, K_i, $i = \mathrm{c}, \mathrm{t}$, is given by

$$K_i^2 = \frac{1}{\sigma^2} \int_{-\infty}^{(\lambda_i-\mu_i)/\sigma} \frac{[g'(s)]^2}{g(s)}\,\mathrm{d}s + \frac{1}{p_i\sigma^2}\left[g\left(\frac{\lambda_i-\mu_i}{\sigma}\right)\right]^2 , \tag{19.6}$$

and λ_i is such that

$$q_i = G\left(\frac{\lambda_i-\mu_i}{\sigma}\right) , \tag{19.7}$$

and $p_i = 1 - q_i$. The required sample size for the treatment group is then $n_\mathrm{t} = \gamma n_\mathrm{c}$. The proof of (19.5) is based on the asymptotic normality of the MLEs of μ_i and can be found in [19.27].

If we want to test the null hypothesis $H_0 : \mu_\mathrm{c} = \mu_\mathrm{t}$ against the one-sided alternative $H_1 : \mu_\mathrm{t} > \mu_\mathrm{c}$ at an asymptotic significance level α $(0 < \alpha < 1)$ and assume that this test is to achieve $100(1-\beta)\%$ power to detect a difference of $d = \mu_\mathrm{t} - \mu_\mathrm{c} > 0$, the required sample size for the control group n_c is given by

$$n_\mathrm{c} = \left(\frac{z_\alpha + z_\beta}{d}\right)^2 \left(\frac{1}{K_\mathrm{c}^2} + \frac{1}{\gamma K_\mathrm{t}^2}\right) , \tag{19.8}$$

and the required sample size for the treatment group is then $n_\mathrm{t} = \gamma n_\mathrm{c}$. The proof of (19.8) can also be found in [19.27].

19.1.2 Examples

Since the family of lognormal distributions and the family of Weibull distributions are two important location-scale families of log-transformed lifetime distributions, we apply our method to these two families.

Example 1: Lognormal Distribution

We first study the family of lognormal distribution in which

$$g(s) = \frac{1}{\sqrt{2\pi}}\,\mathrm{e}^{-s^2/2} . \tag{19.9}$$

For $i = \mathrm{c}, \mathrm{t}$, using integration by parts, we find

$$K_i^2 = -\frac{(\lambda_i-\mu_i)}{\sigma^3} g\left(\frac{\lambda_i-\mu_i}{\sigma}\right) + \frac{q_i}{\sigma^2} + \frac{1}{p_i\sigma^2}\left[g\left(\frac{\lambda_i-\mu_i}{\sigma}\right)\right]^2 , \tag{19.10}$$

where λ_i is such that

$$q_i = \Phi\left(\frac{\lambda_i-\mu_i}{\sigma}\right) . \tag{19.11}$$

If $q_\mathrm{c} = q_\mathrm{t} = 50\%$, then

$$K_i^2 = \frac{1}{2\sigma^2} + \frac{1}{\pi\sigma^2} . \tag{19.12}$$

Example 2: Weibull Distribution

In the family of Weibull distributions,

$$g(s) = \exp(s - \mathrm{e}^s) . \tag{19.13}$$

For $i = \mathrm{c}, \mathrm{t}$, by repeatedly using the technique of integration by parts, we have

$$K_i^2 = \frac{1-p_i}{\sigma^2} . \tag{19.14}$$

If $q_\mathrm{c} = q_\mathrm{t} = 50\%$, then

$$K_i^2 = \frac{1}{2\sigma^2} . \tag{19.15}$$

19.1.3 Effect of Guarantee Time on Sample Size Determination

A very simple feature of lifetime distributions is the existence of a threshold time, or *guarantee time*, during which no subjects will die. For example, the type of mice to be used in the aging intervention testing program of the National Institute on Aging exhibit a guarantee survival time of about 500 days in the survival distribution, as estimated from the survival curves reported by *Turturro* et al. [19.28]. When the comparison of two mean lifetimes is in terms of the difference and when the two distributions share the same guarantee time, this time contributes nothing to the comparison. When the comparison of two mean lifetimes is in terms of the ratio, however, the guarantee time plays an important role in the comparison, especially in the determination of sample sizes at the design stage of the clinical trials.

When the primary concern in a medical study is to compare the means or certain percentiles of two lifetime distributions with type II censored observations such as the aging intervention testing program from the National Institute on Aging, the sample size determination may be based on the method of *Rubinstein* et al. [19.18], the method of *Freedman* [19.22], and the method described by (19.5) and (19.8). The methods of *Rubinstein* et al. [19.18] can be used since the hazard ratio is simply

the reciprocal of the ratio of two means under the exponential distributions. The method of *Freedman* [19.22] refers to the logrank test of the hazard ratio and requires the assumptions of proportional hazards between two groups. It could be used to compare the means or certain percentiles of two lifetime distributions as long as the comparison of means or certain percentiles can be related to the hazard ratio between two distributions such as in the family of Weibull distributions. The method described by (19.5) and (19.8) directly applies to the comparison of means or certain percentiles of two lifetime distributions and requires the assumption of location-scale family of log-transformed lifetime distributions.

If two lifetime distributions under study exhibit survival thresholds, or *guarantee times* as in the survival distribution of mice used in the aging intervention testing program from the National Institute on Aging, these thresholds have an important role in the ratio of means from two distributions. Mathematically, let T_c be the survival time for the control group, and T_t be the survival time for the treatment group. Let ET_c and ET_t be the corresponding means of the two distributions. We are interested in testing the null hypothesis $H_0 : ET_t/ET_c = 1$ against the alternative $H_1 : ET_t/ET_c \neq 1$ at an asymptotic significance level α $(0 < \alpha < 1)$. For $i = c, t$, suppose that T_i follows a distribution with a threshold, or a *guarantee time* $\psi_i > 0$. We assume that both the control group and the treatment group share the same threshold parameter $\psi_1 = \psi_2 = \psi$ and that ψ is known. Let $T_i' = T_i - \psi$, then $ET_i' = ET_i - \psi, i = t, c$. The alternative hypothesis on which the sample size computation is based is

$$\rho = \frac{ET_t}{ET_c} = \frac{ET_t' + \psi}{ET_c' + \psi} . \tag{19.16}$$

Therefore

$$\frac{ET_t'}{ET_c'} = \frac{\rho ET_c - \psi}{ET_c - \psi} . \tag{19.17}$$

Since $ET_t/ET_c = 1$ if and only if $ET_t'/ET_c' = 1$. The original null and alternative hypotheses translate into the corresponding hypotheses in terms of distributions T_t' and T_c' : $H_0' : ET_t'/ET_c' = 1$ and $H_1' : ET_t'/ET_c' \neq 1$.

Since the distributions of T_t' and T_c' begin with time 0 and have no guarantee times, the sample size methods reviewed above can be directly applied to test the reduced hypotheses H_0' against H_1'. The alternative hypothesis on which the sample size computation should be based, however, now becomes $ET_t'/ET_c' = (\rho ET_c - \psi)/(ET_c - \psi)$. We call ET_t'/ET_c' the adjusted effect size. Note that for any $\rho \neq 1$, $ET_t'/ET_c' = \rho$ if and only if $\psi = 0$. Hence, when the ratio of the mean between two lifetime distributions is to be tested, it is crucial that the sample size determination based on the logrank test and the proportional hazards assumption or the location-scale family of log-transformed lifetime distributions takes into account the possible guarantee time in the lifetime distributions. Table 19.1 presents the sample size computation based on the method of *Rubinstein* et al. [19.18] for a selected set of the guarantee time ψ and the percentage p_t of censorship for the treatment group. Table 19.2 presents the sample size computation based on the method of *Freedman* [19.22] for the same selected set of ψ and p_t. The computation in Table 19.2 assumes Weibull distributions with the same shape parameter of 1.5 for both the treatment group and the control group so that the ratio of two means can be expressed as a function of the ratio of hazard functions between the two groups. Table 19.3 presents the sample size computation based on the method described by (19.8) for the same selected set of ψ and p_t under the assumption of lognormal distributions with a scale parameter of 0.8 in the log-transformed lifetime distribution. Table 19.4 presents the sample size computation based on the method described by (19.8) for the same selected set of ψ and p_t under the assumption of Weibull distributions with a scale parameter of 0.8 in the log-transformed lifetime distribution. All these computations in four tables are based on a one-sided test for

Table 19.1 Sample size per group based on the method of *Rubinstein*, et al. [19.18] $\alpha = 5\%$, $\beta = 20\%$

ψ	$p_t = 40\%$	$p_t = 50\%$	$p_t = 60\%$
0	589	702	871
10	518	617	765
20	452	537	666
30	390	463	574
40	332	394	489
50	279	331	410
60	231	274	339
70	187	222	274
80	148	175	216
90	114	134	165
100	84	98	121
110	58	68	83
120	37	43	53
130	21	24	29
140	9	10	12

the ratio of two means at a significance level of 5%, a mean lifetime of 150 units for the control group, and a statistical power of 80%. The 80% power is assumed at $\rho = 1.2$ in the original alternative hypotheses with the equal sample size between the treatment group and the control group. In addition, both groups are assumed the simultaneous entry to the study and the simultaneous stopping time, which is the time of achieving censorship p_t for the treatment group. The censorship for the control group is then decided by ρ and p_t under the appropriate distributional assumptions and is less than p_t by the assumption that $\rho > 1$. The sample size when there is no *guarantee time* or the *guarantee time* is ignored is given when $\psi = 0$.

Although Tables 19.1–19.4 are based on different sample size determination methods, they demonstrate two common important observations. First, if a guarantee time exists in the two lifetime distributions to be compared, the ignorance of the guarantee time leads to the overestimation of the sample size. Second, if the significance level of the test, the statistical power of the test, and the degree of censoring are fixed, the required sample size decreases as the guarantee time increases. All these can be explained by (19.17), which expresses the ratio of two means ET_t'/ET_c' after the guarantee time ψ is subtracted (i. e., the adjusted effect size) as a function of the guarantee time ψ, the original ratio ρ of mean lifetime when the guarantee time is included, and the mean lifetime ET_c for the control group. Let $\rho > 1$ and ET_c be fixed. We denote the adjusted effect size [the right-hand side of (19.17)] by $h(\psi)$. This function has two important features. First, $h(\psi) = \rho$ if and only if $\psi = 0$. Second, the derivative of $h(\psi)$ with respect to ψ is always positive, which implies that it is an increasing function of ψ. Since the sample size methods are applied to the lifetime distributions when the guarantee time is subtracted, the effect size used in these sample size computations is based on $h(\psi)$ instead of ρ. The fact that $h(\psi) = \rho$ if and only if $\psi = 0$ implies that the ig-

Table 19.2 Sample size per group based on the method of *Freedman* [19.22] (Weibull distribution with a shape parameter 1.5 assumed) $\alpha = 5\%$, $\beta = 20\%$

ψ	$p_t = 40\%$	$p_t = 50\%$	$p_t = 60\%$
0	258	305	377
10	227	268	331
20	198	233	288
30	171	201	248
40	146	171	210
50	122	144	176
60	101	119	145
70	83	96	118
80	66	76	93
90	51	59	71
100	38	44	52
110	27	31	36
120	18	20	24
130	12	13	14
140	7	7	8

Table 19.3 Sample size per group based on (19.8); The lognormal case $\alpha = 5\%$, $\beta = 20\%$, $\sigma = 0.8$

ψ	$p_t = 40\%$	$p_t = 50\%$	$p_t = 60\%$
0	267	283	307
10	235	249	270
20	205	217	235
30	178	188	203
40	152	161	174
50	128	135	146
60	106	112	121
70	87	91	99
80	69	73	78
90	53	56	60
100	40	42	45
110	28	29	31
120	18	19	20
130	11	11	12
140	5	5	5

Table 19.4 Sample size per group based on (19.8); the Weibull case $\alpha = 5\%$, $\beta = 20\%$, $\sigma = 0.8$

ψ	$p_t = 40\%$	$p_t = 50\%$	$p_t = 60\%$
0	377	449	558
10	332	395	490
20	289	344	427
30	249	297	368
40	213	253	313
50	179	212	263
60	148	175	217
70	120	142	175
80	95	112	138
90	73	86	106
100	54	63	77
110	37	44	53
120	24	28	34
130	14	16	19
140	6	7	8

norance of the guarantee time (i. e., by assuming $\psi = 0$) will always lead to an inadequate sample size when in fact $\psi > 0$. Since a common feature of the sample size methods is that the sample size decreases when the effect size $h(\psi)$ increases, this explains why the sample size decreases as the guarantee time ψ increases from Tables 19.1 to 19.4.

19.1.4 Application to NIA Aging Intervention Testing Program

We now demonstrate the sample size determination by applying it to the aging intervention testing program (RFA-AG-02-005) of the National Institute on Aging (NIA). One of the major research objectives of this program is to identify interventions that increase mean life expectancy by 10% in phase 1 studies which may be terminated at 50% survivorship. The experimental units in this study are the 4WCNIA mice obtained from the National Institute of Health (NIH) aging rodent colony. Pilot data such as the survival curves reported by *Turturro* et al. [19.28] on similar mice have suggested a guarantee survival time of about 500 days in the survival distribution. In addition, *Pugh* et al. [19.29] reported a mean life expectancy of 876 days and a standard deviation of 18 days for similar mice.

Assume that we ignore the guarantee time in the sample size computation and use $\rho = 1.1$ as the ratio of mean lifetime between the intervention group and the control group. The method of *Rubinstein* et al. [19.18] gives a sample size of 2637 per group. The method of *Freedman* [19.22] gives a sample size of 323 per group based on two Weibull distributions with the same shape parameter which is estimated as 2.793 by the survival curves reported by *Turturro* et al. [19.28]. Our proposed method gives a sample size of 387 per group based on two lognormal distributions with the same scale parameter σ (in the log-transformed lifetime). This computation uses $\sigma = 0.482$ as estimated by the survival curves reported in [19.28]. When applied under the family of Weibull distributions, the projected sample size per group based on our method is 338.

When the 500-days guarantee survival time is taken into account in the sample size computation, these methods are applied to the survival distributions after the 500-days guarantee survival time is subtracted. The pilot information of $ET_c = 876$ and $\rho = 1.1$ along with (19.17) implies that $ET_t'/ET_c' = 1.233$. The sample size methods are then applied to the distributions of T_t' and T_c' when testing H_0' against H_1' at a 5% significance level and an 80% statistical power. We assume that both groups use the same number of mice, that the treatment group is terminated at the 50% censorship, and that the control group is allowed to continue until the treatment group terminates. The method of *Rubinstein* et al. [19.18] gives a sample size of 528 per group. The method of *Freedman* [19.22] under the assumption of Weibull distributions gives a sample size of 64 per group. Assuming the lognormal distribution for the lifetimes with the same scale parameter σ in the log-transformed lifetime distributions between the control and treatment groups, our proposed method gives the sample size required per group as 81. Assuming a Weibull distribution for the lifetimes with the same scale parameter σ in the log-transformed lifetime distributions between the control and treatment groups, our proposed method gives the sample size required per group as 68.

Similar to observations from Tables 19.1–19.4, the real-life example again demonstrates the importance of taking into account the guarantee time in sample size computation when it exists. A considerable waste of resources would occur if the guarantee time is ignored in the sample size projection. Notice also that, while the methods of *Freedman* [19.22] and ours give fairly consistent results about the sample size per group, the method of *Rubinstein* et al. [19.18], however, results in a very different sample size compared to the others. The reason behind this difference is the assumption of an exponential distribution for the method in [19.18]. The mathematically attractive but practically unrealistic property of the exponential distribution is its constant hazard function over time, which then implies the memoryless feature for the survival distribution [19.3]. Although the exponential distribution is a distribution extensively discussed in the fields of biometrics, reliability and industrial life testing literature [19.30, 31], it has long been pointed out by many authors such as *Zelen* and *Dannemiller* [19.32] that the estimations and inferences associated with an exponential distribution are not robust and that exponential distribution is a very unrealistic distribution in many applications, especially in studies associated with the aging process.

19.2 Analysis of Survival Data from Special Cases of Step-Stress Life Tests

We now discuss some applications of survival analysis in engineering and reliability studies. Accelerated life tests (ALT) consist of a variety of test methods for shortening the life of products or hastening the degradation of their performance. The aim of such testing is to obtain data quickly which, properly modeled and analyzed, yield desired information on product life or performance under normal use. ALT can be carried out using constant stress, step-stress, or linearly increasing stress. The step-stress scheme applies stress to test units in the way that the stress setting of test units will be changed at specified times. Generally, a test unit starts at a specified low stress. If the unit does not fail at a specified time, the stress on it is raised and held for a specified time. The stress is repeatedly increased and held, until the test unit fails or a censoring time is reached. A simple step-stress ALT (SSALT) uses only two stress levels. The problem of modeling data from ALT and making inferences from such data has been studied by many authors. *Chernoff* [19.33] considered optimal life tests for estimation of model parameters based on data from ALT. *Meeker* and *Nelson* [19.34] obtained optimum ALT plans for Weibull and extreme-value distributions with censored data. *Nelson* and *Kielpinski* [19.35] further studied optimum ALT plans for normal and lognormal life distributions based on censored data. *Nelson* [19.36] considered data from SSALT and obtained maximum likelihood estimates (MLE) for the parameters of a Weibull distribution under the inverse power law using the breakdown time data of an electrical insulation. *Miller* and *Nelson* [19.37] studied optimum test plans which minimized the asymptotic variance of the MLE of the mean life at a design stress for simple SSALT where all units were run to failure. *Bai* et al. [19.38] further studied the similar optimum simple SSALT plan for the case where a specified censoring time was involved. *Tyoskin* and *Krivolapov* [19.39] presented a nonparametric approach for making inferences for SSALT data. *Dorp* et al. [19.40] developed a Bayes model and studied the inferences of data from SSALT. *Xiong* [19.41] obtained inferences based on pivotal quantities for type II censored exponential data from a simple SSALT. *Alhadeed* and *Yang* [19.42] discussed the optimal simple step-stress plan for the Khamis–Higgins model. *Teng* and *Yeo* [19.43] used the method of least squares to estimate the life–stress relationship in SSALT. *Hobbs* [19.44] gave detailed discussion on highly accelerated life test (HALT) and highly accelerated stress screens (HASS). *Mann* et al. [19.45] and *Lawless* [19.3] provided the general theory and applications of lifetime data analysis. *Meeker* and *Escobar* [19.46] briefly surveyed optimum test plans for different types of ALT. *Nelson* [19.1, 47] provided an extensive and comprehensive source for theory and examples for ALT and SSALT.

During the step-stress life test, test units can be continuously or intermittently inspected for failure. The latter type of test is frequently used since it generally requires less testing effort and can be administratively more convenient. In some other cases, intermittent inspection is the only feasible way of checking the status of test units (see, for example, [19.48]). The data obtained from intermittent inspections are called grouped data and consist of only the number of failures in the inspection intervals. The first problem we consider in this section is the statistical inference of model parameters and optimum test plans based on only grouped and type I censored data obtained from a step-stress life test. We will also study another important and interesting variation associated with grouped and censored data from a simple SSALT, when both the stress change time and the censoring time are random variables, such as the order statistics at the current stress levels, and when only these order statistics (stress-change time and type II censoring time) are observed during the test.

Throughout the section, we denote the design stress by x_0, the i-th test stress by x_i, $i = 1, 2, \cdots, m$, $x_1 < x_2 < \cdots < x_m$, where m is the total number of test stress levels. We assume that the i-th stress change time is constant τ_i, $i = 1, 2, \cdots, m-1$, and the fixed censoring time is $\tau_m > \tau_{m-1}$. Let $\tau_{m+1} = \infty$, $\tau_0 = 0$, $\Delta\tau_i = \tau_i - \tau_{i-1}$. We also make following assumptions:

(A1). At any constant stress x_i, $i = 0, 1, 2, \cdots, m$, the cumulative distribution function (CDF)of a test unit lifetime is

$$F_i(t) = F(t/\theta_i) \quad \text{for} \quad t > 0\,, \tag{19.18}$$

where the stress–response relationship (SRR) θ_i is a function of stress x_i and F is a strictly increasing distribution function.

(A2). The stresses are applied in the order $x_1 < x_2 < \cdots < x_m$.

(A3). The lifetimes of test units under SSALT are statistically independent.

For the step-stress life test, there is a probability distribution $G(t)$ of time T to failure on test. Data from this distribution are observed during the test. The cumulative exposure model of time T assumes that the remaining

life of a test unit depends only on the current cumulative fraction failed and the current stress, regardless of how the fraction is accumulated. Moreover, if held at the current stress, survivors will fail according to the cumulative distribution for that stress but starting at the previously accumulated fraction failed. Also, the change in stress has no effect on life, only the level of the stress does. As pointed out by *Miller* and *Nelson* [19.37] and *Yin* and *Sheng* [19.49], this model has many applications in industrial life testing.

Mathematically, the cumulative distribution $G(t)$ of time T to failure from a step-stress test described above is

$$G(t) = \begin{cases} F_i(t - \tau_{i-1} + s_{i-1}) \,, \\ \quad \text{for} \quad \tau_{i-1} \le t < \tau_i \,, \\ \qquad\qquad i = 1, 2, \cdots, m-1, \\ F_m(t - \tau_{m-1} + s_{m-1}) \,, \\ \quad \text{for} \quad \tau_{m-1} \le t < \infty \,, \end{cases} \tag{19.19}$$

where s_{i-1} is the equivalent start time at step i satisfying

$$F_i(s_{i-1}) = F_{i-1}(\tau_{i-1} - \tau_{i-2} + s_{i-2}) \,. \tag{19.20}$$

We further assume:

(A4). The stress-response relationship (SRR) $\theta_i = \theta(x_i)$ is a log–linear function of stress x_i. That is,

$$\log \theta(x_i) = \alpha + \beta x_i \,, \tag{19.21}$$

where α and β are unknown model parameters which typically depend on the nature of the product and the test method. Although the specification of $\theta(x)$ looks rather restrictive, it covers some of the most important models used in industry, such as the power-law model, the Eyring model and the Arrhenius model [19.47]. With the above specifications, it is straightforward to find that $s_{i-1} = (\tau_{i-1} - \tau_{i-2} + s_{i-2})\theta_i/\theta_{i-1} = \theta_i \sum_{j=1}^{i-1} \Delta\tau_j/\theta_j$, $i = 2, 3, 4, \cdots, m$. Thus, the distribution function of the step-stress failure time T is

$$G(t) = \begin{cases} F\left(\dfrac{t - \tau_{i-1}}{\theta_i} + \sum_{j=1}^{i-1} \dfrac{\Delta\tau_j}{\theta_j}\right) \,, \\ \quad \text{for } \tau_{i-1} \le t < \tau_i, \, i = 1, 2, ..., m-1 \,. \\ F\left(\dfrac{t - \tau_{m-1}}{\theta_m} + \sum_{j=1}^{m-1} \dfrac{\Delta\tau_j}{\theta_j}\right) \,, \\ \quad \text{for } \tau_{m-1} \le t < \infty \end{cases} \tag{19.22}$$

Although we will discuss the statistical analysis of data collected from SSALT in the context of engineering studies, we point out that the statistical models and methods discussed in this section can be readily used in medical and biological research. Many clinical trials on therapeutical compounds of a disease contain a novel treatment group and a control group. At the completion of well-designed clinical studies, especially when the novel treatment has been found efficacious for the protection against the disease development based on the available data, many clinical trials are extended for another specified time period so that subjects from the original control group can receive the treatment. When a survival endpoint such as the time from the study baseline to the onset of a specific disease is measured on subjects from the original control group throughout the entire trial period, the resulting survival data are very analogous to these collected from the standard SSALT in engineering studies. In fact, the drug dose used in such clinical trials can be thought of as the stress level in which 0 is the stress at the initial phase of the original control group and a positive dose is the stress at the extension phase of the trial.

19.2.1 Analysis of Grouped and Censored Data from Step-Stress Life Tests

MLE

We first consider the case when all test units are subject to the same censoring time and the same stress-change patterns with the same set of stresses and the same stress-change times. We assume that data obtained from such step-stress tests are grouped and type I censored. More specifically, we assume that the intermittent inspection times during the step-stress test coincide with the stress-change times and the censoring time. Suppose that n test units begin at low stress x_1. If the unit does not fail at a specified time τ_1, the stress on it is raised to a higher stress x_2. The stress is repeatedly increased and held in this fashion, until the test unit fails or the fixed censoring time τ_m ($> \tau_{m-1}$) is reached. Assume that n_i units fail during the inspection time interval $[\tau_{i-1}, \tau_i)$, $i = 1, 2, \cdots, m, m+1$, ($\tau_{m+1} = \infty$). To simplify the notations, we denote, for $i = 1, 2, \cdots, m+1$,

$$\begin{aligned} u_i(\alpha, \beta) &= \sum_{j=1}^{i} \Delta\tau_j \, \exp(-\alpha - \beta x_j) \\ v_i(\alpha, \beta) &= \sum_{j=1}^{i} x_j \Delta\tau_j \, \exp(-\alpha - \beta x_j) \,. \end{aligned} \tag{19.23}$$

Let $p_i = \Pr(\tau_{i-1} \le T < \tau_i)$ for $1 \le i \le m+1$. The cumulative exposure model (19.22) implies that, for

$1 \le i \le m+1$,

$$p_i = F[u_i(\alpha,\beta)] - F[u_{i-1}(\alpha,\beta)] \,. \tag{19.24}$$

The likelihood function based on data vector $(n_1, n_2, \cdots, n_{m+1})$ is (up to a constant):

$$L(\alpha,\beta) \propto \prod_{i=1}^{m+1} \{F[u_i(\alpha,\beta)] - F[u_{i-1}(\alpha,\beta)]\}^{n_i} \,, \tag{19.25}$$

where θ_i is specified by the SRR (19.21) and $F(\infty) = 1$. Thus, the log likelihood function is a function of the unknown parameters α and β :

$$\log L(\alpha,\beta) \propto \sum_{i=1}^{m+1} n_i \log \{F[u_i(\alpha,\beta)] - F[u_{i-1}(\alpha,\beta)]\} \,. \tag{19.26}$$

To find the maximum likelihood estimators (MLE) for α and β, we maximize $\log L(\alpha,\beta)$ over α and β. The maximization of $\log L(\alpha,\beta)$ requires the solution to the system:

$$\begin{cases} \dfrac{\partial L(\alpha,\beta)}{\partial \alpha} = -\sum_{i=1}^{m+1} n_i \times \dfrac{u_i(\alpha,\beta) f[u_i(\alpha,\beta)]}{F[u_i(\alpha,\beta)] - F[u_{i-1}(\alpha,\beta)]} - \dfrac{u_{i-1}(\alpha,\beta) f[u_{i-1}(\alpha,\beta)]}{F[u_i(\alpha,\beta)] - F[u_{i-1}(\alpha,\beta)]} = 0 \,, \\ \dfrac{\partial L(\alpha,\beta)}{\partial \beta} = -\sum_{i=1}^{m+1} n_i \times \dfrac{v_i(\alpha,\beta) f[u_i(\alpha,\beta)]}{F[u_i(\alpha,\beta)] - F(u_{i-1}(\alpha,\beta))} - \dfrac{v_{i-1}(\alpha,\beta) f[u_{i-1}(\alpha,\beta)]}{F[u_i(\alpha,\beta)] - F(u_{i-1}(\alpha,\beta))} = 0 \,, \end{cases} \tag{19.27}$$

where $f(t) = \mathrm{d}F(t)/\mathrm{d}t$ is the probability density function of $F(t)$. Generally, the solution of (19.27) requires a numerical method such as Newton–Raphson. *Seo* and *Yum* [19.50] proposed several approximate ML estimators and compared with the MLE by a Monte Carlo simulation when the lifetime distribution is assumed exponential. The expected second partial derivatives of the log likelihood function at $(\alpha,\beta)'$ are

$$\begin{aligned} \sigma_{11} &= -E\frac{\partial^2 L(\alpha,\beta)}{\partial \alpha^2} \\ &= -n\sum_{i=1}^{m+1} \frac{\partial^2 p_i}{\partial \alpha^2} + \sum_{i=1}^{m+1} \frac{1}{p_i}\left(\frac{\partial p_i}{\partial \alpha}\right)^2 \,, \\ \sigma_{12} &= -E\frac{\partial^2 L(\alpha,\beta)}{\partial \alpha \partial \beta} \\ &= -n\sum_{i=1}^{m+1} \frac{\partial^2 p_i}{\partial \alpha \partial \beta} + \sum_{i=1}^{m+1} \frac{1}{p_i}\frac{\partial p_i}{\partial \alpha}\frac{\partial p_i}{\partial \beta} \,, \\ \sigma_{22} &= -E\frac{\partial^2 L(\alpha,\beta)}{\partial \beta^2} \\ &= -n\sum_{i=1}^{m+1} \frac{\partial^2 p_i}{\partial \beta^2} + \sum_{i=1}^{m+1} \frac{1}{p_i}\left(\frac{\partial p_i}{\partial \beta}\right)^2 \,. \end{aligned} \tag{19.28}$$

Let D be the $(m+1)$ by $(m+1)$ diagonal matrix with $1/p_i$, $i = 1, 2, \cdots, m+1$, as its diagonal elements. Let $J = (j_{st})$, $1 \le s \le m+1$, $1 \le t \le 2$, be the $(m+1)\times 2$ Jacobian matrix of $(p_1, p_2, \cdots, p_{m+1})'$ with respect to $(\alpha,\beta)'$, i. e.,

$$\begin{aligned} j_{s1} = \frac{\partial p_i}{\partial \alpha} &= -u_i(\alpha,\beta) f[u_i(\alpha,\beta)] \\ &\quad + u_{i-1}(\alpha,\beta) f[u_{i-1}(\alpha,\beta)] \,, \\ j_{s2} = \frac{\partial p_i}{\partial \beta} &= -v_i(\alpha,\beta) f[u_i(\alpha,\beta)] \\ &\quad + v_{i-1}(\alpha,\beta) f[u_{i-1}(\alpha,\beta)] \,, \end{aligned} \tag{19.29}$$

for $s = 1, 2, \cdots, m+1$. Because

$$E\left(\sum_{i=1}^{m+1} \frac{\partial p_i}{\partial \alpha}\right) = E\left(\sum_{i=1}^{m+1} \frac{\partial p_i}{\partial \beta}\right) = 0 \,,$$

the expected Fisher information matrix $\boldsymbol{\Sigma} = (\sigma_{ij})$, $i, j = 1, 2$, is given by $\boldsymbol{\Sigma} = n \cdot J' D J$. Let $(\widehat{\alpha}, \widehat{\beta})'$ be the MLE of $(\alpha,\beta)'$ obtained from solving (19.27). $n^{-1}\boldsymbol{\Sigma}$ can be consistently estimated by $n^{-1}\widehat{\boldsymbol{\Sigma}}$, where $\widehat{\boldsymbol{\Sigma}} = (\widehat{\sigma_{ij}})$, $i, j = 1, 2$, is obtained by replacing $(\alpha,\beta)'$ in $\boldsymbol{\Sigma}$ by its MLE $(\widehat{\alpha}, \widehat{\beta})'$.

Based on the asymptotic normality of $(\widehat{\alpha}, \widehat{\beta})'$ with estimated covariance matrix $\widehat{\Sigma}^{-1}$, we can set up the asymptotic confidence interval (CI) for α, β, the SRR of lifetime at design stress $\theta_0 = \theta(x_0) = \exp(\alpha + \beta x_0)$, and the reliability function at design stress $R_0(t) = 1 - F(t/\theta_0)$. Let $\widehat{\Sigma}^{-1} = (\widehat{m_{ij}})$, $i, j = 1, 2$, be the estimated asymptotic covariance matrix of $(\widehat{\alpha}, \widehat{\beta})'$. It is straightforward to show that an asymptotic $100(1-\gamma)\%$ CI for α is $\widehat{\alpha} \pm z_{\gamma/2}\widehat{m}_{11}$, and an asymptotic $100(1-$

$\gamma)\%$ CI for β is $\widehat{\beta} \pm z_{\gamma/2}\widehat{m}_{22}$, where $z_{\gamma/2}$ is the $\gamma/2$ point of the standard normal distribution. The asymptotic variance for $\log\widehat{\theta(x_0)} = \widehat{\alpha} + \widehat{\beta}x_0$ is given by

$$\widehat{\sigma} = (1, x_0)\widehat{\boldsymbol{\Sigma}}^{-1}(1, x_0)' \,. \tag{19.30}$$

An asymptotic $100(1-\gamma)\%$ CI for θ_0 is

$$\exp\left(\widehat{\alpha} + \widehat{\beta}x_0 \pm z_{\gamma/2}\widehat{\sigma}\right) \,.$$

Finally, because $F(t)$ is a strictly increasing function of t, an asymptotic $100(1-\gamma)\%$ CI for $R_0(t) = 1 - F(t/\theta_0)$ at a given time t is $1 - F\left[t/\exp\left(\widehat{\alpha} + \widehat{\beta}x_0 \pm z_{\gamma/2}\widehat{\sigma}\right)\right]$.

When the step-stress test is a simple SSALT, i. e., when $m = 2$, there exist closed form MLE for α and β. The MLE of α and β solves

$$\begin{cases} F\left(\dfrac{\tau_1}{\theta_1}\right) & = \dfrac{n_1}{n} \\ F\left(\dfrac{\tau_2 - \tau_1}{\theta_2} + \frac{\tau_1}{\theta_1}\right) - F\left(\dfrac{\tau_1}{\theta_1}\right) & = \dfrac{n_2}{n} \,. \end{cases} \tag{19.31}$$

The solutions are

$$\begin{aligned} \widehat{\alpha} &= \frac{x_2}{x_2 - x_1}\log\frac{\tau_1}{F^{-1}\left(\frac{n_1}{n}\right)} \\ &\quad - \frac{x_1}{x_2 - x_1}\log\frac{(\tau_2 - \tau_1)}{F^{-1}\left(\frac{n_1+n_2}{n}\right) - F^{-1}\left(\frac{n_1}{n}\right)} \,, \\ \widehat{\beta} &= \frac{1}{x_2 - x_1}\log\frac{(\tau_2 - \tau_1)F^{-1}\left(\frac{n_1}{n}\right)}{\tau_1\left[F^{-1}\left(\frac{n_1+n_2}{n}\right) - F^{-1}\left(\frac{n_1}{n}\right)\right]} \,, \end{aligned} \tag{19.32}$$

where F^{-1} is the inverse function of F.

In the more general situation when different test units are subject to different censoring times and different stress-change patterns with different sets of stresses and even different stress-change times, the likelihood function for each test unit can be given by (19.25) for each test unit with $\sum_{i=1}^{m+1} n_i = 1$. The likelihood function for a sample of size n test units is the product of all n individual likelihood functions by assumption (A3). The MLE of α and β can be obtained by maximizing this likelihood function using a numerical method such as Newton–Raphson. Although the lifetime distributions of n test units are independent, they are not identical. The asymptotic confidence interval estimates for various model parameters given above, however, are still valid when the Fisher information matrix $\boldsymbol{\Sigma}/n$ is replaced by the average information matrix to take into account of the difference in the lifetime distributions. The detailed theoretical justification can be found in Chapt. 9 of *Cox* and *Hinkley* [19.51].

A Statistical Test for the Cumulative Exposure Model when m>2

We only consider the case when all test units are subject to the same censoring time and the same stress-change patterns with the same set of stresses and the same stress-change times in this section. We again let $p_i = \Pr(\tau_{i-1} \le T < \tau_i)$ for $1 \le i \le m+1$. The cumulative exposure model (19.22) implies that

$$p_i = F[u_i(\alpha, \beta)] - F[u_{i-1}(\alpha, \beta)] \,. \tag{19.33}$$

A statistical test for the cumulative exposure model can be obtained by testing the null hypothesis $H_0: \; p_i = F[u_i(\alpha, \beta)] - F[u_{i-1}(\alpha, \beta)], \; 1 \le i \le m+1$, against the alternative H_a : there is no constraint on p_i, $1 \le i \le m+1$. When grouped and type I censored data are available from n test units, the likelihood function of p_i, $1 \le i \le m+1$, is

$$L \propto \prod_{i=1}^{m+1} {p_i}^{n_i} \,. \tag{19.34}$$

The MLE of p_i, $1 \le i \le m+1$, under H_0 are given by

$$\widehat{p_i^0} = F\left[u_i\left(\widehat{\alpha}, \widehat{\beta}\right)\right] - F\left[u_{i-1}\left(\widehat{\alpha}, \widehat{\beta}\right)\right] \,, \tag{19.35}$$

where $\widehat{\alpha}, \widehat{\beta}$ are the MLE of α and β. Under H_a, a straightforward maximization of the likelihood function gives the MLE of p_i, $1 \le i \le m+1$, as

$$\widehat{p_i^a} = \frac{n_i}{n} \,. \tag{19.36}$$

Therefore, an asymptotic likelihood ratio test of significance level γ $(0 < \gamma < 1)$ rejects H_0 if

$$\begin{aligned} &-2\sum_{i=1}^{m+1} n_i \left(\log\left\{F\left[u_i\left(\widehat{\alpha}, \widehat{\beta}\right)\right]\right. \right. \\ &\left. \left. - \; F\left[u_{i-1}\left(\widehat{\alpha}, \widehat{\beta}\right)\right]\right\} - \log\frac{n_i}{n}\right) \\ &> \chi^2_\gamma(m-2) \,, \end{aligned} \tag{19.37}$$

where $\chi^2_\gamma(m-2)$ is the upper $100\gamma\%$ percentile of the χ^2 distribution with $m-2$ degrees of freedom. Because

$$\begin{aligned} &-2\sum_{i=1}^{m+1} n_i \left(\log\left\{F\left[u_i\left(\widehat{\alpha}, \widehat{\beta}\right)\right]\right. \right. \\ &\left. \left. - \; F\left[u_{i-1}\left(\widehat{\alpha}, \widehat{\beta}\right)\right]\right\} - \log\frac{n_i}{n}\right) \end{aligned} \tag{19.38}$$

is stochastically equivalent to

$$\sum_{i=1}^{m+1} \frac{\left(n_i - n\left\{F\left[u_i\left(\widehat{\alpha},\widehat{\beta}\right)\right] - F\left[u_{i-1}\left(\widehat{\alpha},\widehat{\beta}\right)\right]\right\}\right)^2}{n\left\{F\left[u_i\left(\widehat{\alpha},\widehat{\beta}\right)\right] - F\left[u_{i-1}\left(\widehat{\alpha},\widehat{\beta}\right)\right]\right\}}, \tag{19.39}$$

another asymptotically equivalent test of significance level γ $(0 < \gamma < 1)$ is the well-known χ^2 goodness-of-fit test, which rejects H_0 if

$$\sum_{i=1}^{m+1} \frac{\left(n_i - n\left\{F\left[u_i\left(\widehat{\alpha},\widehat{\beta}\right)\right] - F\left[u_{i-1}\left(\widehat{\alpha},\widehat{\beta}\right)\right]\right\}\right)^2}{n\left\{F\left[u_i\left(\widehat{\alpha},\widehat{\beta}\right)\right] - F\left[u_{i-1}\left(\widehat{\alpha},\widehat{\beta}\right)\right]\right\}} > \chi^2_\gamma(m-2)\,. \tag{19.40}$$

The mathematical verification of these tests can be found in *Agresti* [19.52] and *Pearson* [19.53].

Optimum Test Plans

We next discuss the optimum test plan for choosing τ_1 in a particular case. Suppose that n test units are tested under a simple SSALT which uses the same censoring time τ_2 and the same stress-change patterns with the same set of stresses $(x_1 < x_2)$ and the same stress-change times τ_1. Assume that the censoring time τ_2 is given. Suppose that the lifetimes at constant stresses x_1 and x_2 are exponential with means θ_1 and θ_2, respectively, where $\theta_i = \exp(\alpha + \beta x_i)$, $i = 1, 2$. Thus, $F(t) = 1 - \exp(-t)$ for $t > 0$. The expected Fisher information matrix $\boldsymbol{\Sigma}$ is now simplified as

$$\boldsymbol{\Sigma} = n\begin{pmatrix} A\left(\frac{\tau_1}{\theta_1}\right)^2 + B\left(\frac{\Delta\tau_2}{\theta_2}\right)^2 & x_1 A\left(\frac{\tau_1}{\theta_1}\right)^2 + x_2 B\left(\frac{\Delta\tau_2}{\theta_2}\right)^2 \\ x_1 A\left(\frac{\tau_1}{\theta_1}\right)^2 + x_2 B\left(\frac{\Delta\tau_2}{\theta_2}\right)^2 & A\left(\frac{x_1\tau_1}{\theta_1}\right)^2 + B\left(\frac{x_2\Delta\tau_2}{\theta_2}\right)^2 \end{pmatrix}, \tag{19.41}$$

where

$$A = (1-p_1)p_1$$
$$B = (1-p_1)/[\exp(\Delta\tau_2/\theta_2) - 1]\,.$$

Because

$$\boldsymbol{\Sigma}^{-1} = \frac{(\theta_1\theta_2)^2}{n \cdot AB\tau_1^2\Delta\tau_2^2(x_2-x_1)^2} \times \begin{pmatrix} A\left(\frac{x_1\tau_1}{\theta_1}\right)^2 + B\left(\frac{x_2\Delta\tau_2}{\theta_2}\right)^2 & -x_1 A\left(\frac{\tau_1}{\theta_1}\right)^2 - x_2 B\left(\frac{\Delta\tau_2}{\theta_2}\right)^2 \\ -x_1 A\left(\frac{\tau_1}{\theta_1}\right)^2 - x_2 B\left(\frac{\Delta\tau_2}{\theta_2}\right)^2 & A\left(\frac{\tau_1}{\theta_1}\right)^2 + B\left(\frac{\Delta\tau_2}{\theta_2}\right)^2 \end{pmatrix}, \tag{19.42}$$

we find that the asymptotic variance of $\log\widehat{\theta}_0 = \widehat{\alpha} + \widehat{\beta}x_0$, denoted by $\mathrm{Asvar}\left(\log\widehat{\theta}_0\right)$, is given by

$$n \cdot \mathrm{Asvar}\left(\log\widehat{\theta}_0\right) = \xi^2 \frac{\theta_2^2\left[\exp(\Delta\tau_2/\theta_2) - 1\right]}{\exp(-\tau_1/\theta_1)\,(\Delta\tau_2)^2} + (1+\xi)^2 \times \frac{\theta_1^2\left[1 - \exp(-\tau_1/\theta_1)\right]}{\exp(-\tau_1/\theta_1)\tau_1^2}, \tag{19.43}$$

where $\xi = \frac{x_1 - x_0}{x_2 - x_1}$ is the amount of stress extrapolation. Our optimum criterion is to find the optimum stress change time τ_1 $(0 < \tau_1 < \tau_2)$ such that the $\mathrm{Asvar}(\log\widehat{\theta}_0)$ is minimized. Because

$$\lim_{\tau_1 \to 0^+} \mathrm{Asvar}\left(\log\widehat{\theta}_0\right) = \lim_{\tau_1 \to \tau_2^-} \mathrm{Asvar}\left(\log\widehat{\theta}_0\right) = +\infty\,, \tag{19.44}$$

the minimum of $\mathrm{Asvar}\left(\log\widehat{\theta}_0\right)$ is attained at some τ_1 between 0 and τ_2 based on the fact that $\mathrm{Asvar}\left(\log\widehat{\theta}_0\right)$ is a continuous function of τ_1 when τ_1 is between 0 and τ_2. The minimization of $\mathrm{Asvar}\left(\log\widehat{\theta}_0\right)$ over τ_1 solves the equation

$$\frac{\partial\left[n \cdot \mathrm{Asvar}\left(\log\widehat{\theta}_0\right)\right]}{\partial\tau_1} = 0\,, \tag{19.45}$$

where

$$\frac{\partial\left[n \cdot \mathrm{Asvar}\left(\log\widehat{\theta}_0\right)\right]}{\partial\tau_1} = \frac{\xi^2\theta_2^2}{(\Delta\tau_2)^3}\exp\left(\frac{\tau_1}{\theta_1}\right)\left\{\left[2 + \left(\frac{1}{\theta_1} - \frac{1}{\theta_2}\right)\Delta\tau_2\right]\exp\left(\frac{\Delta\tau_2}{\theta_2}\right) - \left(2 + \frac{\Delta\tau_2}{\theta_1}\right)\right\} + \frac{(1+\xi)^2\theta_1^2}{\tau_1^3}\left[2 + \left(\frac{\tau_1}{\theta_1} - 2\right)\exp\left(\frac{\tau_1}{\theta_1}\right)\right]. \tag{19.46}$$

The uniqueness of the solution to (19.45) is shown in [19.54]. In general, the solution to (19.45) is not in a closed form and therefore requires a numerical method such as the Newton–Raphson method.

An Example

We use a real data set reported in Table 17.2.1 of Chapt. 10 in *Nelson* [19.47] to demonstrate our estimation and testing procedure. The data set was obtained from a step-stress test of cryogenic cable insulation. Each specimen was first stressed for 10 min each at voltages of 5 kV, 10 kV, 15 kV, and 20 kV before it went into

step 5. Thereafter one group of specimens was stressed for 15 min at each step given in Table 19.5.

Three other groups were held for 60, 240, and 960 min at each step. Thus there were four step-stress patterns. The stress on a specimen (x) is the natural logarithm of the ratio between the voltage and the insulation thickness.

The original data were observed as exact failure times. To demonstrate our estimation process, we grouped the failure time data according to the intervals formed by consecutive stress-change times. There were five censored failure times in the data set. The grouped and censored data are summarized in Table 19.6.

Because of the different thickness for different specimens and different voltages at different steps in the testing, each specimen has its own stress pattern and censoring time. A likelihood function can be written for each specimen according to (19.25). The likelihood function for the whole sample is the product of all these individual likelihood functions in the sample. By using exact failure times instead of grouped count data, *Nelson* [19.47] fitted the Weibull model to the step-stress data and presented the MLE of model parameters on Table 17.2.2 of Chapt. 10 in *Nelson* [19.47]. The MLE estimate for the Weibull shape parameter is 0.755 97 with an asymptotic 95% confidence interval from 0.18 to 1.33. Because the confidence interval contains the value 1, there is no significant evidence against the hypothesis that the failure times of these specimens follow an exponential distribution when tested against the larger family of Weibull distributions based on the standard normal test at a significance level of 5%. We choose to base our analysis on exponential failure time in the step-stress test.

The analysis provided by Chapt. 10 in *Nelson* [19.47] assumed that the SRR is an inverse power-law model and used the stress as the ratio between the voltage and the insulation thickness. In our set up of log-linear SRR, the inverse power-law model translates into $\log\theta(x) = \alpha + \beta x$, where $\theta(x)$ is the mean of the exponential distribution at stress x, and stress x now becomes the natural logarithm of the ratio between the voltage and the insulation thickness. The design stress is at 400 V/mm, therefore, $x_0 = 5.99$. Table 19.7 presents the MLE and CI of various parameters.

To demonstrate how to find the optimum design under a simple step-stress life test, we assume that the voltage levels from step 5 (26 kV) and step 6

Table 19.5 Step-stress pattern after step 4

Step	5	6	7	8	9	10	11
kV	26.0	28.5	31.0	33.4	36.0	38.5	41.0

Table 19.6 Count data

Holding time (min)	Final step	Count (uncensored)	Censoring time	Count (censored)	Thickness (mm)
15	9	3		0	27
60	10	1	370	1	29.5
60	10	0	345	1	28
240	9	2		0	29
240	10	2	1333	1	29
240	10	1		0	30
960	5	1		0	30
960	5	0	363.9	1	30
960	6	1		0	30
960	7	3	2460.9	1	30
960	8	1		0	30
960	9	1		0	30

Table 19.7 Parameter estimates

Parameter	MLE	95% CI
α	97.5	[60.3, 134.7]
β	-12.9	$[-13.6, -12.2]$
$\theta(x_0) = \exp(\alpha + \beta x_0)$	6.1×10^{8}	$[2.05\times10^{-8}, 1.76\times10^{9}]$
$R_0(t) = \exp[-t/\theta(x_0)]$	$\exp(-10^{-8}t/6.1)$	$[\exp(-10^{-8}t/2.05), \exp(-10^{-9}t/1.76)]$

(28.5 kV) in the step-stress pattern are used to conduct a future simple step-stress life test. We also assume that the test uses the cable insulation with thickness equal to 30 mm (one of the four types used in the study). Therefore, the two stress levels for this simple step-stress test are $x_1 = \log(26\,000/30) = 6.765$ and $x_2 = \log(28\,500/30) = 6.856$. We still use a design stress of $x_0 = 5.99$. The amount of stress extrapolation is $\xi = 8.516$. We assume that the simple step-stress test has to stop after 1800 min (censoring time τ_2). Using the MLE of α and β obtained from the grouped and censored data in Table 19.7, we numerically solved (19.45) and found that the optimum stress-change time is after 1191.6 min (τ_1) of testing under stress x_1.

19.2.2 Analysis of a Very Simple Step-Stress Life Test with a Random Stress-Change Time

In this section we deal with a very special case of a simple SSALT that is subject to type II censoring. The traditional cumulative exposure model assumes that the stress-change time is a prespecified constant. The stress-change time in many applications, however, can be a random variable which follows a distribution. Here we consider a specific case of a simple step-stress life testing in which the stress-change time T_1 is an order statistic at the low-stress level. This type of simple SSALT occurs when experimenters want to change the stress level after a certain number of failures are observed at the low-stress level.

Throughout the section, we also denote the design stress by x_0, the i-th test stress by x_i, $i = 1, 2$, $x_1 < x_2$. We further assume that a sample of n test units begin at the low stress x_1 until the first n_1 units fail. The stress is then raised to the high stress x_2 and held until another n_2 units fail. Let $C = n(n-n_1)\binom{n-n_1-1}{n_2-1}\binom{n-1}{n_1-1}$ and $\binom{n}{k} = n!/[k!(n-k)!]$ for $0 \le k \le n$. For $0 \le i \le n_1 - 1$ and $0 \le j \le n_2 - 1$, let $\xi(n, n_1, i) = n - n_1 + i + 1$ and $\eta(n, n_1, n_2, j) = n - n_1 - n_2 + j + 1$. In addition to the assumption (A1) and (A2) made above, we further assume that only two order statistics are observed during the entire simple SSALT: one is the stress-change time, which is the n_1-th order statistic under the low stress x_1, the other is the final failure time of SSALT, which is the n_2-th order statistic under the high stress x_2. We will present the joint and marginal distributions of the two observed order statistics from the simple SSALT. We will also discuss the maximum likelihood estimates (MLE) and the method of moment estimates (MME) for the model parameters based on the joint distribution and present the exact confidence interval estimates for the model parameters based on various pivotal quantities.

Joint Distribution of Order Statistics under SSALT

Let T_1 be the stress-change time and T be the lifetime under such a simple SSALT. We further assume that the lifetime under the simple SSALT, given $T_1 = t_1$, follows the cumulative exposure model. Therefore, the conditional cumulative distribution function $G_{T|T_1}$ of T, given the stress-change time $T_1 = t_1$, is given by the classic cumulative exposure model [19.47]:

$$G_{T|T_1}(t) \begin{cases} F_1(t), & \text{for} \quad 0 \le t < t_1 \\ F_2(t - \tau_1 + s), & \text{for} \quad t_1 \le t < \infty \end{cases}, \tag{19.47}$$

where $s = t_1\theta_2/\theta_1$. The conditional probability distribution function (PDF) $g(t|t_1)$ of T, given $T_1 = t_1$, is then

$$g(t|t_1) = \begin{cases} \dfrac{1}{\theta_1} f\left(\dfrac{t_1}{\theta_1}\right), & \text{for} \quad 0 \le t < t_1 \\ \dfrac{1}{\theta_2} f\left(\dfrac{t_2 - t_1}{\theta_2} + \dfrac{t_1}{\theta_1}\right), & \text{for} \quad t_1 \le t < \infty \end{cases}. \tag{19.48}$$

The marginal probability density function (PDF) of T is given by $g(t|t_1)l(t_1)$, where $l(t_1)$ is the PDF of T_1 [19.55].

Suppose that T_2 ($T_1 < T_2$) is the final censoring observation under the simple SSALT. The observed data in such a test are the vector (T_1, T_2). Since T_1 is the n_1-th smallest observation from the distribution $F(\frac{t_1}{\theta_1})$. The probability density function of T_1 is [19.3]:

$$l(t_1) = \binom{n-1}{n_1-1} \frac{n}{\theta_1} f\left(\frac{t_1}{\theta_1}\right) F^{n_1-1}\left(\frac{t_1}{\theta_1}\right) \times \left[1 - F\left(\frac{t_1}{\theta_1}\right)\right]^{n-n_1}. \tag{19.49}$$

Given $T_1 = t_1$, the conditional cumulative exposure model implies that T_2 is the n_2-th order statistic from a sample of size $n - n_1$ with probability density function $(1/\theta_2) f[(t_2 - t_1)/\theta_2 + t_1/\theta_1]/[1 - F(t_1/\theta_1)]$, $t \ge t_1$. Thus, the conditional probability density function for T_2,

given $T_1 = t_1$, is [19.3]:

$$
\begin{aligned}
f_{T_2|T_1}(t_2) = & \binom{n-n_1-1}{n_2-1} \frac{(n-n_1)}{\theta_2} \\
& \times f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) \\
& \times \left[F\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) - F\left(\frac{t_1}{\theta_1}\right)\right]^{n_2-1} \\
& \times R^{n-n_1-n_2}\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) \\
& \times R^{n_1-n}\left(\frac{t_1}{\theta_1}\right) , \qquad (19.50)
\end{aligned}
$$

where $R(t) = 1 - F(t)$. Therefore, the joint probability density for (T_1, T_2) is

$$
\begin{aligned}
f(t_1, t_2) &= f_{T_2|T_1}(t_2) l_1(t_1) \\
&= \frac{C}{\theta_1\theta_2} f\left(\frac{t_1}{\theta_1}\right) f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) \\
&\quad \times F^{n_1-1}\left(\frac{t_1}{\theta_1}\right) \\
&\quad \times \left[F\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) - F\left(\frac{t_1}{\theta_1}\right)\right]^{n_2-1} \\
&\quad \times R^{n-n_1-n_2}\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) . \qquad (19.51)
\end{aligned}
$$

When the lifetime is exponential under constant stress, i. e., $f(t) = \exp(-t)$, $t > 0$,

$$
\begin{aligned}
f(t_1, t_2) = & \frac{C}{\theta_1\theta_2}\left\{\exp\left[-(n-n_1+1)\frac{t_1}{\theta_1}\right]\right\} \\
& \times \left[1-\exp\left(-\frac{t_1}{\theta_1}\right)\right]^{n_1-1} \\
& \times \left\{\exp\left[-(n-n_1-n_2+1)\frac{t_2-t_1}{\theta_2}\right]\right\} \\
& \times \left[1-\exp\left(-\frac{t_2-t_1}{\theta_2}\right)\right]^{n_2-1} . \qquad (19.52)
\end{aligned}
$$

MLE and MME

From now on we concentrate on the SRR, which assumes that $\theta(x)$ is a log-linear function of the stress x, i. e., $\ln[\theta(x)] = \alpha + \beta x$. The parameters α and β are characteristics of the products and test methods and we assume that $x > 0$ and $\beta < 0$. Notice that $\theta(x)$ is a multiple of the mean lifetime under the stress x based on the assumption (A1). In fact, if the lifetime distribution is exponential, then $\theta(x)$ is the mean lifetime under stress x. We discuss the point estimates for α, β, and $\theta_0 = \exp(\alpha + \beta x_0)$ in this section based on the method of maximum likelihood and the method of moment.

As a function of α and β, the joint density function $f(t_1, t_2)$ in (19.51) becomes the likelihood function $L(\alpha, \beta)$ based on the data vector (T_1, T_2). The maximum likelihood estimate for α and β can be obtained by solving the system of equations:

$$
\begin{cases}
\begin{aligned}
\frac{\partial \log L}{\partial \alpha} &= -2 - \frac{t_1 f'\left(\frac{t_1}{\theta_1}\right)}{\theta_1 f\left(\frac{t_1}{\theta_1}\right)} - \left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) \\
&\quad \times \frac{f'\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)}{f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)} - \frac{(n_1-1)t_1 f\left(\frac{t_1}{\theta_1}\right)}{\theta_1 F\left(\frac{t_1}{\theta_1}\right)} \\
&\quad - (n_2-1) \\
&\quad \times \frac{\left[\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) - \frac{t_1}{\theta_1} f\left(\frac{t_1}{\theta_1}\right)\right]}{F\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) - F\left(\frac{t_1}{\theta_1}\right)} \\
&\quad + (n-n_1-n_2)\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) \\
&\quad \times \frac{f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)}{R\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)} = 0
\end{aligned} \\
\begin{aligned}
\frac{\partial \log L}{\partial \beta} &= -\frac{x_1 t_1 f'\left(\frac{t_1}{\theta_1}\right)}{\theta_1 f\left(\frac{t_1}{\theta_1}\right)} - \left[\frac{x_2(t_2-t_1)}{\theta_2}+\frac{x_1 t_1}{\theta_1}\right] \\
&\quad \times \frac{f'\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)}{f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)} - \frac{(n_1-1)x_1 t_1 f\left(\frac{t_1}{\theta_1}\right)}{\theta_1 F\left(\frac{t_1}{\theta_1}\right)} \\
&\quad - (x_1+x_2) \\
&\quad - (n_2-1) \\
&\quad \times \left[\frac{\left(\frac{x_2(t_2-t_1)}{\theta_2}+\frac{x_1 t_1}{\theta_1}\right) f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)}{F\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) - F\left(\frac{t_1}{\theta_1}\right)} - \frac{\frac{x_1 t_1}{\theta_1} f\left(\frac{t_1}{\theta_1}\right)}{F\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right) - F\left(\frac{t_1}{\theta_1}\right)}\right] \\
&\quad + (n-n_1-n_2)\left[\frac{x_2(t_2-t_1)}{\theta_2}+\frac{x_1 t_1}{\theta_1}\right] \\
&\quad \times \frac{f\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)}{R\left(\frac{t_2-t_1}{\theta_2}+\frac{t_1}{\theta_1}\right)} = 0
\end{aligned}
\end{cases} , \qquad (19.53)
$$

where $f'(t) = \mathrm{d}f(t)/\mathrm{d}t$. In general, the solution of (19.53) requires a numerical method such as the Newton–Raphson method. The methods in *Seo* and *Yum* [19.50] can also be used. To find the MME of α and β, we notice that (by a change of variable)

$$
\begin{aligned}
ET_1 = & \int_0^\infty t_1 l(t_1)\,\mathrm{d}t_1 = n\binom{n-1}{n_1-1}\theta_1 \\
& \times \int_0^1 u^{n_1-1}(1-u)^{n-n_1} F^{-1}(u)\,\mathrm{d}u , \qquad (19.54)
\end{aligned}
$$

and

$$E(T_2|T_1=t_1)=\int_{t_1}^{\infty} t_2 f_{T_2|T_1}(t_2)\,\mathrm{d}t_2$$
$$=\binom{n-n_1-1}{n_2-1}\frac{(n-n_1)}{R^{n-n_1}\left(\frac{t_1}{\theta_1}\right)}$$
$$\times\int_{F(t_1/\theta_1)}^{1}\left[\theta_2 F^{-1}(v)+\left(1-\frac{\theta_2}{\theta_1}\right)t_1\right]$$
$$\times\left[v-F\left(\frac{t_1}{\theta_1}\right)\right]^{n_2-1}(1-v)^{n-n_1-n_2}\,\mathrm{d}v\,, \tag{19.55}$$

where F^{-1} is the inverse function of F. Thus, by letting $w=F(t_1/\theta_1)$,

$$E(T_2)=E_{T_1}\left[E(T_2|T_1)\right]$$
$$=C\int_0^1 w^{n_1-1}\,\mathrm{d}w$$
$$\times\int_w^1\left[\theta_2 F^{-1}(v)+(\theta_1-\theta_2)\,F^{-1}(w)\right]$$
$$\times[v-w]^{n_2-1}(1-v)^{n-n_1-n_2}\,\mathrm{d}v\,. \tag{19.56}$$

The MME of α and β can be found by solving the system of equations:

$$\begin{cases} T_1 &= n\binom{n-1}{n_1-1}\theta_1\int_0^1 u^{n_1-1}(1-u)^{n-n_1}F^{-1}(u)\,\mathrm{d}u \\ T_2 &= C\int_0^1 w^{n_1-1}\,\mathrm{d}w \\ &\quad\times\int_w^1\left[F^{-1}(v)+(\theta_1-\theta_2)\,F^{-1}(w)\right] \\ &\quad\times(v-w)^{n_2-1}(1-v)^{n-n_1-n_2}\,\mathrm{d}v\,. \end{cases} \tag{19.57}$$

When the lifetime is exponential under constant stress, i. e., $f(t)=\exp(-t)$, $t>0$, $ET_1=\theta_1\sum_{i=0}^{n_1-1}(n-i)^{-1}$ [19.3]. A direct binomial expansion in (19.52) along with the repeated use of integration by parts yields

$$E(T_2)=\theta_1\sum_{i=0}^{n_1-1}(n-i)^{-1}$$
$$+C\sum_{i=0}^{n_1-1}\sum_{j=0}^{n_2-1}(-1)^{i+j}\binom{n_1-1}{i}$$
$$\times\binom{n_2-1}{j}\frac{\theta_2}{\eta^2(n,n_1,n_2,j)\xi(n,n_1,i)}\,. \tag{19.58}$$

This also gives a closed form solution to the MME of α and β as

$$\widetilde{\beta}=\frac{1}{x_2-x_1}\left\{\ln\left[(T_2-T_1)\sum_{i=0}^{n_1-1}(n-i)^{-1}\right]\right.$$
$$-\ln\left[T_1 C\sum_{i=0}^{n_1-1}\sum_{j=0}^{n_2-1}(-1)^{i+j}\binom{n_1-1}{i}\binom{n_2-1}{j}\right.$$
$$\left.\left.\times\,\eta^{-2}(n,n_1,n_2,j)\zeta^{-1}(n,n_1,i)\right]\right\}$$
$$\widetilde{\alpha}=\ln\left(\frac{T_1}{\sum_{i=0}^{n_1-1}(n-i)^{-1}}\right)-\widetilde{\beta}x_1\,. \tag{19.59}$$

Confidence Interval Estimates of Model Parameters

Now we set up the exact confidence intervals for α, β, and $\theta_0=\exp(\alpha+\beta x_0)$ under the assumption that F is given. We first observe an important fact which will be used for the estimation of model parameters in this section. Let $S_1=\frac{T_1}{\theta_1}$, and $S_2=\frac{T_2-T_1}{\theta_2}$. The joint probability density function of (S_1, S_2) is

$$g(s_1,s_2)=Cf(s_1)f(s_2+s_1)F^{n_1-1}(s_1)$$
$$\times[F(s_2+s_1)-F(s_1)]^{n_2-1}$$
$$\times[1-F(s_2+s_1)]^{n-n_1-n_2}\,. \tag{19.60}$$

Therefore, (S_1, S_2) is a pivotal vector whose distribution does not depend on the unknown parameters θ_1 and θ_2.

We now set up a confidence interval for β. Let $S_3=\frac{S_2}{S_1}$. The marginal distribution of S_3 is given by

$$g_3(s_3)=\int_0^{\infty} g(s_1,s_1s_3)s_1\,\mathrm{d}s_1\,. \tag{19.61}$$

Since $S_3=\frac{T_2-T_1}{T_1}\exp\left[\beta(x_1-x_2)\right]$, a $100(1-\gamma)\%$ $(0<\gamma<1)$ confidence interval for β is $[\beta_1,\beta_2]$, where

$$\beta_1=\frac{1}{x_2-x_1}\ln\left(\frac{T_2-T_1}{T_1 S_{3,\gamma/2}}\right),$$
$$\beta_2=\frac{1}{x_2-x_1}\ln\left(\frac{T_2-T_1}{T_1 S_{3,1-\gamma/2}}\right), \tag{19.62}$$

and for $0<c<1$, $S_{3,c}$ is such that

$$\int_0^{S_{3,c}} g_3(s_3)\,\mathrm{d}s_3=1-c\,. \tag{19.63}$$

To set up a confidence interval for α, we let $S_4 = S_1^{x_2/x_1}$ and $S_5 = \frac{S_2}{S_4}$. The marginal distribution of S_5 is given by

$$g_5(s_5) = \frac{x_1}{x_2} \int_0^\infty g\left(s_4^{\frac{x_1}{x_2}}, s_4 s_5\right) s_4^{\frac{x_1}{x_2}} \, \mathrm{d}s_4 \,. \tag{19.64}$$

Since $S_5 = (T_2 - T_1)\, T_1^{-\frac{x_2}{x_1}} \exp\left(\frac{x_2 - x_1}{x_1}\alpha\right)$, a $100(1-\gamma)\%$ $(0 < \gamma < 1)$ confidence interval for α is $[\alpha_1, \alpha_2]$, where

$$\begin{aligned} \alpha_1 &= \frac{x_1}{x_1 - x_2}\left[\ln(T_2 - T_1) - \ln\left(T_1^{\frac{x_2}{x_1}} S_{5,1-\gamma/2}\right)\right], \\ \alpha_2 &= \frac{x_1}{x_1 - x_2}\left[\ln(T_2 - T_1) - \ln\left(T_1^{\frac{x_2}{x_1}} S_{5,\gamma/2}\right)\right], \end{aligned} \tag{19.65}$$

and, for $0 < c < 1$, $S_{5,c}$ is such that

$$\int_0^{S_{5,c}} g_5(s_5)\,\mathrm{d}s_5 = 1 - c \,. \tag{19.66}$$

The confidence interval for $\theta_0 = \exp(\alpha + \beta x_0)$ can also be obtained based on the distribution of a similar pivotal quantity to S_5. In fact, for $i = 1, 2$, we can always write $\theta_i = \exp(\alpha + \beta x_i) = \exp[(\alpha + \beta x_0) + \beta(x_i - x_0)]$. Therefore, by replacing the stress x_i by the transformed stress $x_i - x_0$ in the derivation of pivotal quantity S_5, we obtain another pivotal quantity:

$$\widetilde{S}_5 = (T_2 - T_1)\, T_1^{-\frac{x_2 - x_0}{x_1 - x_0}} \exp\left[\frac{x_2 - x_1}{x_1 - x_0}(\alpha + \beta x_0)\right] . \tag{19.67}$$

The distribution of $\widetilde{S}_5$ is given by the marginal density function

$$\widetilde{g}_5(s_5) = \frac{x_1 - x_0}{x_2 - x_0} \int_0^\infty g\left(s_4^{\frac{x_1 - x_0}{x_2 - x_0}}, s_4 s_5\right) s_4^{\frac{x_1 - x_0}{x_2 - x_0}} \, \mathrm{d}s_4 \,. \tag{19.68}$$

By using the distribution of pivotal quantity $\widetilde{S}_5$, we can set up a confidence interval for $\alpha + \beta x_0$ similar to the way that the confidence interval for α was set up based on the original stress x_i and the pivotal quantity S_5. Then a confidence interval for θ_0 can be obtained by the exponentiation of the confidence interval of $\alpha + \beta x_0$. More specifically, let $\xi = (x_1 - x_0)/(x_2 - x_1)$ be the amount of stress extrapolation [19.37]. A $100(1-\gamma)\%$ $(0 < \gamma < 1)$ confidence interval for $\theta_0 = \exp(\alpha + \beta x_0)$ is $[\theta_{01}, \theta_{02}]$, where

$$\begin{aligned} \theta_{01} &= \frac{T_1^{1+\xi}\, \widetilde{S}_{5,1-\gamma/2}^{\xi}}{(T_2 - T_1)^{\xi}} \,, \\ \theta_{02} &= \frac{T_1^{1+\xi}\, \widetilde{S}_{5,\gamma/2}^{\xi}}{(T_2 - T_1)^{\xi}} \,, \end{aligned} \tag{19.69}$$

Table 19.8 Percentiles of S_3 and S_5

Variable	Percentile	$n_2 = 6$	$n_2 = 8$	$n_2 = 10$
S_3	1	0.40	0.69	1.07
	2.5	0.52	0.86	1.31
	5	0.63	1.03	1.55
	10	0.80	1.27	1.88
	90	4.01	5.82	8.20
	95	5.12	7.38	10.36
	97.5	6.37	9.14	12.79
	99	8.28	11.83	16.52
S_5	1	0.84	1.32	1.91
	2.5	1.08	1.73	2.57
	5	1.42	2.25	3.32
	10	1.97	3.07	4.51
	90	27.14	40.05	56.98
	95	42.16	61.96	87.96
	97.5	62.95	92.25	130.75
	99	102.81	150.25	212.60

and, for $0 < c < 1$, $\widetilde{S}_{5,c}$ is such that

$$\int_0^{\widetilde{S}_{5,c}} \widetilde{g}_5(s_5)\,\mathrm{d}s_5 = 1 - c\,. \tag{19.70}$$

When the lifetime is exponential under constant stress, i. e., $f(t) = \exp(-t)$, $t > 0$, the marginal density functions for S_3 and S_5 are simplified. The marginal probability density function of S_3 is

$$g_3(s_3) = C \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} (-1)^{i+j} \binom{n_1-1}{i} \binom{n_2-1}{j} \times \frac{1}{[\xi(n, n_1, i) + \eta(n, n_1, n_2, j)s_3]^2}\,. \tag{19.71}$$

A direct integration gives the marginal CDF of S_3 as

$$G_3(s_3) = C \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} (-1)^{i+j} \binom{n_1-1}{i} \binom{n_2-1}{j} \times \frac{1}{\eta(n, n_1, n_2, j)} \left[\frac{1}{\xi(n, n_1, i)} - \frac{1}{[\xi(n, n_1, i) + \eta(n, n_1, n_2, j)s_3]} \right]. \tag{19.72}$$

The marginal density function of S_5 is

$$g_5(s_5) = C \frac{x_1}{x_2} \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} (-1)^{i+j} \binom{n_1-1}{i} \binom{n_2-1}{j} \times \int_0^{\infty} \exp\left[-\xi(n, n_1, i) s_4^{\frac{x_1}{x_2}} - \eta(n, n_1, n_2, j) s_4 s_5 \right] s_4^{\frac{x_1}{x_2}}\,\mathrm{d}s_4\,. \tag{19.73}$$

A change in the order of integration and the use of integration by parts gives the marginal CDF of S_5 as

$$G_5(s_5) = C \frac{x_1}{x_2} \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} (-1)^{i+j} \binom{n_1-1}{i} \binom{n_2-1}{j} \times \int_0^{\infty} \frac{1 - \exp[-\eta(n, n_1, n_2, j)s_4, s_5]}{\eta(n, n_1, n_2, j)s_4} \times \exp\left[-\zeta(n, n_1, i) s_4^{\frac{x_1}{x_2}} \right] s_4^{\frac{x_1}{x_2}}\,\mathrm{d}s_4 \tag{19.74}$$

The marginal density function and the marginal distribution function of $\widetilde{S}_5$ can be obtained by replacing the stress x_i by the transformed stress $x_i - x_0$, $i = 1, 2$, in the corresponding function of S_5. Except for trivial situations, numerical integration subroutines are typically required for the evaluation of the distribution functions associated with the pivotal quantities even when the exponential distributions are assumed. In addition, the approximation of these distributions can also be obtained through large simulations of the pivotal quantities.

Assume that a sample of 20 experimental units are placed under a simple step-stress life test. The test stress is changed from the lower stress x_1 to the higher stress x_2 after the fifth failure from the lower stress level x_1 is observed ($n_1 = 5$). The test is finished after another n_2 failures are observed at the higher stress x_2. Assume that the lifetime distribution under constant stress $x_i (i = 0, 1, 2)$ is exponential with mean parameter $\theta_i = \exp(\alpha + \beta x_i)$. For $x_0 = 0, x_2 = 2x_1 > 0$ and $n_2 = 6, 8, 10$, Table 19.8 presents the 1, 2.5, 5, 10, 90, 95, 97.5 and 99 percentiles for the distributions of S_3 and S_5. These percentiles are computed by numerical integration of the distribution function for S_3 and S_5. They can be used to set up appropriate confidence intervals for β, α, and θ_0.

References

19.1 W. B. Nelson: *Applied Life Data Analysis* (Wiley, New York 1982)

19.2 J. D. Kalbfleisch, R. L. Prentice: *The Statistical Analysis of Failure Time Data* (Wiley, New York 1980)

19.3 J. F. Lawless: *Statistical Models and Methods for Lifetime Data* (Wiley, New York 1982)

19.4 R. Peto, M. C. Pike, P. Armitage, N. E. Breslow, D. R. Cox, S. V. Howard, N. Mantel, K. McPherson, J. Peto, P. G. Smith: Design and analysis of randomized clinical trials requiring prolonged observation of each patient. Part II: Analysis and examples, Br. J. Cancer **35**, 1–39 (1977)

19.5 D. R. Cox: Regression models and life tables (with Discussion), J. R. Stat. Soc. B **74**, 187–200 (1972)

19.6 D. Schoenfeld: Partial residuals for the proportional hazards regression model, Biometrika **69**, 239–241 (1982)

19.7 T. M. Therneau, P. M. Grambsch, T. R. Fleming: Martingale-based residuals and survival models, Biometrika **77**, 147–160 (1990)

19.8 T. M. Therneau, P. M. Grambsch: *Modeling Survival Data: Extending the Cox Model* (Springer, Berlin Heidelberg New York 2000)

19.9 T. R. Fleming, D. P. Harrington: *Counting Processes and Survival Analysis* (Wiley, New York 1991)

19.10 P. K. Anderson, R. D. Gill: Cox's regression model for counting processes: a large sample study, Ann. Stat. **10**, 1100–1120 (1982)

19.11 P. Tiraboschi, L. A. Hansen, E. Masliah, M. Alford, L. J. Thal, J. Corey-Bloom: Impact of APOE genotype on neuropathologic and neurochemical markers of Alzheimer disease, Neurology **62(11)**, 1977–1983 (2004)

19.12 R. Weindruch, R. L. Walford: *The Retardation of Aging and Disease by Dietary Restriction* (Thomas, Springfield 1988)

19.13 H. M. Brown-Borg, K. E. Borg, C. J. Meliska, A. Bartke: Dwarf mice and the aging process, Nature **33**, 384 (1996)

19.14 R. A. Miller: Kleemeier award lecture: are there genes for aging?, J Gerontol. **54A**, B297–B307 (1999)

19.15 H. R. Warner, D. Ingram, R. A. Miller, N. L. Nadon, A. G. Richardson: Program for testing biological interventions to promote healthy aging., Mech. Aging Dev. **155**, 199–208 (2000)

19.16 S. L. George, M. M. Desu: Planning the size and duration of a clinical trial studying the time to some critical event, J. Chron. Dis. **27**, 15–24 (1974)

19.17 D. A. Schoenfeld, J. R. Richter: Nomograms for calculating the number of patients needed for aclinical trial with survival as an endpoint, Biometrics **38**, 163–170 (1982)

19.18 L. V. Rubinstein, M. H. Gail, T. J. Santner: Planning the duration of acomparative clinical trial with loss to follow-up and a period of continued observation, J. Chron. Dis. **34**, 469–479 (1981)

19.19 J. Halperin, B. W. Brown: Designing clinical trials with arbitrary specification of survival functions and for the log rank or generalized Wilcoxon test, Control. Clin. Trials **8**, 177–189 (1987)

19.20 E. Lakatos: Sample sizes for clinical trials with time-dependent rates of losses and noncompliance, Control. Clin. Trials **7**, 189–199 (1986)

19.21 D. Schoenfeld: The asymptotic properties of nonparametric tests for comparing survival distributions, Biometrika **68**, 316–318 (1981)

19.22 L. S. Freedman: Tables of the number of patients required in clinical trials using the log-rank test, Stat. Med. **1**, 121–129 (1982)

19.23 E. Lakatos: Sample sizes based on the log-rank statistic in complex clinical trials, Biometrics **44**, 229–241 (1988)

19.24 M. Wu, M. Fisher, D. DeMets: Sample sizes for long-term medical trial with time-dependent noncompliance and event rates, Control. Clin. Trials **1**, 109–121 (1980)

19.25 E. Lakatos, K. K. G. Lan: A comparison of sample size methods for the logrank statistic, Stat. Med. **11**, 179–191 (1992)

19.26 J. Crowley, D. R. Thomas: *Large Sample Theory for the Log Rank Test*, Technical Report, Vol. 415 (University of Wisconsin, Department of Statistics, 1975)

19.27 C. Xiong, Y. Yan, M. Ji: Sample sizes for comparing means of two lifetime distributions with type II censored data: application in an aging intervention study, Control. Clin. Trials **24**, 283–293 (2003)

19.28 A. Turturro, W. W. Witt, S. Lewis et al.: Growth curves and survival characteristics of the animals used in the biomarkers of aging program, J. Gerontol. Biol. Sci. Med. Sci. **A54**, B492–B501 (1999)

19.29 T. D. Pugh, T. D. Oberley, R. I. Weindruch: Dietary intervention at middle age: caloric restriction but not dehydroepiandrosterone sulfate increases lifespan and lifetime cancer incidence in mice, Cancer Res. **59**, 1642–1648 (1999)

19.30 A. S. Little: Estimation of the T-year survival rate from follow-up studies over alimited period of time, Human Biol. **24**, 87–116 (1952)

19.31 B. Epstein: Truncated life tests in the exponential case, Ann. Math. Stat. **23**, 555–564 (1954)

19.32 M. Zelen, M. C. Dannemiller: The robustness of life testing procedures derived from the exponential distribution, Technometrics **3**, 29–49 (1961)

19.33 H. Chernoff: Optimal accelerated life designs for estimation, Technometrics **4**, 381–408 (1962)

19.34 W. Q. Meeker, W. B. Nelson: Optimum accelerated life tests for Weibull and extreme value distributions and censored data, IEEE Trans. Reliab. **24**, 321–332 (1975)

19.35 W. B. Nelson, T. J. Kielpinski: Theory for optimum censored accelerated life tests for normal and lognormal life distributions, Technometrics **18**, 105–114 (1976)

19.36 W. B. Nelson: Accelerated life testing—step-stress models and data analysis, IEEE Trans. Reliab. **29**, 103–108 (1980)

19.37 R. W. Miller, W. B. Nelson: Optimum simple step-stress plans for accelerated life testing, IEEE Trans. Reliab. **32**, 59–65 (1983)

19.38 D. S. Bai, M. S. Kim, S. H. Lee: Optimum simple step-stress accelerated life tests with censoring, IEEE Trans. Reliab. **38**, 528–532 (1989)

19.39 O. I. Tyoskin, S. Y. Krivolapov: Nonparametric model for step-stress accelerated life test, IEEE Trans. Reliab. **45**, 346–350 (1996)

19.40 J. R. Dorp, T. A. Mazzuchi, G. E. Fornell, L. R. Pollock: A Bayes approach to step-stress accelerated life test, IEEE Trans. Reliab. **45**, 491–498 (1996)

19.41 C. Xiong: Inferences on a simple step-stress model with type II censored exponential data, IEEE Trans. Reliab. **47**, 142–146 (1998)

19.42 A. A. Alhadeed, S. S. Yang: Optimal simple step-stress plan for Khamis–Higgins model, IEEE Trans. Reliab. **51**, 212–215 (2002)
19.43 S. L. Teng, K. P. Yeo: A least-square approach to analyzing life–stress relationship in step-stress accelerated life tests, IEEE Trans. Reliab. **51**, 177–182 (2002)
19.44 G. K. Hobbs: *Accelerated Reliability Engineering* (Wiley, New York 2000)
19.45 N. R. Mann, R. E. Schafer, N. D. Singpurwalla: *Methods for Statistical Analysis of Reliability and Life Data* (Wiley, New York 1974)
19.46 W. Q. Meeker, L. A. Escobar: A review of recent research, current issues in accelerated testing, **61**, 147–168 (1993)
19.47 W. B. Nelson: *Accelerated Life Testing, Statistical Models, Test Plans, and Data Analysis* (Wiley, New York 1990)
19.48 S. Ehrenfeld: Some experimental design problems in attribute life testing, J. Am. Stat. Assoc. **57**, 668–679 (1962)
19.49 X. K. Yin, B. Z. Sheng: Some aspects of accelerated life testing by progressive stress, IEEE Trans. Reliab. **36**, 150–155 (1987)
19.50 S. K. Seo, B. J. Yum: Estimation methods for the mean of the exponential distribution based on grouped censored data, IEEE Trans. Reliab. **42**, 87–96 (1993)
19.51 D. R. Cox, D. V. Hinkley: *Theoretical Statistics* (Chapman Hall, London 1974)
19.52 A. Agresti: *Categorical Data Analysis* (Wiley, New York 1990)
19.53 K. Pearson: On a criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling, Philos. Mag. **50**, 157–175 (1900)
19.54 C. Xiong, M. Ji: Analysis of grouped and censored data from step-stress life testing, IEEE Trans. Reliab. **53**(1), 22–28 (2004)
19.55 C. Xiong: Step-stress life-testing with random stress-change times for exponential data, IEEE Trans. Reliab. **48**, 141–148 (1999)

20. Failure Rates in Heterogeneous Populations

Most of the papers on failure rate modeling deal with homogeneous populations. Mixtures of distributions present an effective tool for modeling heterogeneity. In this chapter we consider nonasymptotic and asymptotic properties of mixture failure rates in different settings.

After a short introduction, in the first section of this chapter we show (under rather general assumptions) that the mixture failure rate is 'bent-down' compared with the corresponding unconditional expectation of the baseline failure rate, which has been proved in the literature for some specific cases. This property is due to an effect where 'the weakest populations die out first', explicitly proved mathematically in this section. This should be taken into account when analyzing failure data for heterogeneous populations in practice. We also consider the problem of mixture failure rate ordering for the ordered mixing distributions. Two types of stochastic ordering are analyzed: ordering in the likelihood ratio sense and ordering the variances when the means are equal. Mixing distributions with equal expectations and different variances can lead to corresponding ordering for mixture failure rates in $[0,\infty)$ in some specific cases. For a general mixing distribution, however, this ordering is only guaranteed for sufficiently small t.

In the second section, the concept of proportional hazards (PH) in a homogeneous population is generalized to a heterogeneous case. For each subpopulation, the PH model is assumed to exist. It is shown that this proportionality is violated for observed (mixture) failure rates. The corresponding bounds for a mixture failure rate are obtained in this case. The change point in the environment is discussed. Shocks – changing the mixing distribution – are also considered. It is shown that shocks with the stochastic properties described also bend down the initial mixture failure rate.

Finally, the third section is devoted to new results on the asymptotic behavior of mixture failure rates. The suggested lifetime model generalizes all three conventional survival models (proportional hazards, additive hazards and accelerated life) and makes it possible to derive explicit asymptotic results. Some of the results obtained can be generalized to a wider class of lifetime distributions, but it appears that the class considered is 'optimal' in terms of the trade-off between the complexity of a model and the tractability (or applicability) of the results. It is shown that the mixture failure rate asymptotic behavior depends only on the behavior of a mixing distribution near to zero, and not on the whole mixing distribution.

Although most studies that model failure rates deal with homogeneous cases, homogeneous populations are rare in real life. Neglecting the existence of heterogeneity can lead to substantial errors during stochastic analy-

sis, reliability, survival and risk analysis, and in other disciplines.

Mixtures of distributions usually present an effective approach to modeling heterogeneity. There may be a physical origin for such mixing in practice. This may happen, for instance, if different (heterogeneous) types of devices that all perform the same function, and are not distinguishable during operation, are mixed together. This occurs in real life when we have 'identical' items that originate from different brands. A similar situation arises when data from different distributions are pooled to enlarge the sample size.

It is well-known that mixtures of decreasing failure rate (DFR) distributions are always also DFR [20.1]. On the other hand, mixtures of increasing failure rate distributions (IFR) can decrease, at least over some intervals of time, which means that the IFR class of distributions is not closed under the operation of mixing [20.2]. As IFR distributions are usually used to model lifetimes governed by aging processes, this means that the operation of mixing can change the pattern of aging dramatically; for example from positive aging (IFR) to negative aging (DFR). It should be noted, however, that the change in the aging pattern usually occurs at sufficiently large item age, and so asymptotic methods are clearly important in this type of analysis. These facts and other implications of heterogeneity should be taken into account in applications.

One specific natural approach to this modeling exploits a notion of a non-negative random unobserved parameter (the frailty) Z, introduced by *Vaupel* et al. [20.3] for a gamma-distributed Z. This, in fact, can be interpreted as a subjective approach and leads to a consideration of a random failure rate $\lambda(t, Z)$. Some interesting applications of the frailty concept in survival analysis were studied by *Aalen* [20.4]. Since the failure rate is a conditional characteristic, the 'ordinary' expectation $E[\lambda(t, Z)]$ with respect to Z does not define a mixture failure rate $\lambda_m(t)$, and a proper conditioning should be performed [20.5]. It is worth mentioning that a random failure rate is a specific case of a hazard rate process [*Kebir* [20.6] and *Yashin* and *Manton* [20.7]]. A convincing 'experiment' that shows a deceleration in the observed failure rate is performed by nature. It is well-known that human mortality follows the *Gompertz* [20.8] lifetime distribution with an exponentially increasing mortality rate. Assume that heterogeneity can be described by the proportional gamma frailty model:

$$\lambda(t, Z) = Z\alpha \exp(\beta t),$$

where α and β are positive constants. Due to its computational simplicity, the gamma frailty model is practically the only one that has been used in applications so far. It can be shown (see, e.g., *Finkelstein* and *Esaulova* [20.9]) that the mixture failure rate $\lambda_m(t)$ in this case is monotonic in $[0, \infty)$ and asymptotically tends to a constant as $t \to \infty$. However, $\lambda_m(t)$ monotonically increases for real values of the parameters of this model. This fact explains the recently observed deceleration in human mortality for the oldest humans (human mortality plateau, as in *Thatcher* [20.10]). A similar result has been experimentally obtained for a large cohort of medflies by *Carey* et al. [20.11]. On the other hand, in engineering applications a mixing operation can result in a failure rate that increases for $[0, t_m)$, $t_m > 0$ and decreases asymptotically to 0 for (t_m, ∞), which has been experimentally observed by *Finkelstein* [20.12] for example for a heterogeneous sample of miniature light bulbs (Example 20.1). This fact is easily explained theoretically using the gamma frailty model with a baseline failure rate that increases as a power function (Weibull law) [20.9, 13].

When considering heterogeneous populations in different environments, the problem of ordering mixture failure rates for stochastically ordered random mixing variables arises. This topic has not been addressed in the literature before. In Sect. 20.1 we show that the natural type of ordering for the mixing models under consideration is ordering by likelihood ratio [20.14, 15]. This correlates with the general considerations of *Block* et al. [20.16] with respect to burn-in of heterogeneous populations. Specifically, when two frailties are ordered in this way, the corresponding mixture failure rates are naturally ordered as functions of time in $[0, \infty)$. Some specific results for the case of frailties with equal means and different variances are also obtained.

In Sect. 20.2 we discuss a 'combination' of a frailty and a proportional hazards (PH) model. The case of a step-stress change-point in the proportional hazards framework is considered and the corresponding bounds for the mixture failure rate are also obtained. Another example deals with a special type of shock, which performs a burn-in for heterogeneous populations.

Section 20.3 is devoted to the important topic of the asymptotic behavior of mixture failure rates. In *Block* et al. [20.17], it was proved that, if the failure rate of each subpopulation converges to a constant and this convergence is uniform, then the mixture failure rate converges to the failure rate of the strongest subpopulation: in other words, the weakest subpopulations die out

first. (For convenience, from now on we shall use, where appropriate, the term "population" instead of "subpopulation") This result is a generalization of the case where populations have constant failure rates, as considered by *Clarotti* and *Spizzichino* [20.18], and it also represents a further development of the work by *Block* et al. in [20.16] (see also [20.19,20]). In *Block* and *Joe* [20.21] the following asymptotic result, which addresses the issue of ultimate monotonicity, was obtained. Let z_0 be a realization of a frailty Z, which corresponds to the strongest population. If $\lambda(t, z)/\lambda(t, z_0)$ uniformly decreases as $t \to \infty$, then $\lambda_m(t)/\lambda(t, z_0)$ also decreases. If, in addition, $\lim_{t\to\infty} \lambda(t, z_0)$ exists, then this quotient decreases to 1. Although the lifetime model obtained from these findings may be rather general, the analytical restrictions, such as uniform convergence, are rather stringent. Besides, the strongest population cannot always be identified.

We suggest a class of distributions that generalizes the proportional hazards, the additive hazards and the accelerated life models and we prove a simple asymptotic result for the mixture failure rate for this class of lifetime distributions. It turns out that the asymptotic behavior of mixture failure rates depends only on the behavior of the mixing distribution in the neighborhood of the left end point of its support, and not on the whole mixing distribution.

Notation

The following notation is used in this chapter

T	lifetime random variable,
$F(t)$	cumulative distribution function of T,
Z	unobserved random variable (frailty),
$F(t, z)$	cumulative distribution function indexed by parameter z,
$\Pi(z)$	distribution function of Z,
$\Pi(z\|t)$	conditional distribution function of Z,
$\pi(z)$	probability density function of Z,
$\pi_k(z)$	probability density function of kZ,
$\pi(z\|t)$	conditional probability density of Z,
$\lambda(t, z)$	failure rate indexed by parameter z,
$\Lambda(t, z)$	cumulative failure rate indexed by parameter z,
$\lambda_m(t)$	mixture failure rate,
$\lambda_P(t)$	unconditional expectation in the family of failure rates,
$\lambda_{mk}(t)$	mixture failure rate for the PH model,
$\tilde{\lambda}_{mk}(t)$	notation for $k\lambda_m(t)$,
$\lambda_{ms}(t)$	mixture failure rate after a shock,
$g(z)$	function decreasing in z
$\varepsilon(t)$	baseline stress,
$\varepsilon_s(t)$	more severe stress,
$A(s)$	function defining the general survival model,
$\phi(t)$	scale function in the general survival model,
$\Psi(t)$	additive part of the general survival model.

20.1 Mixture Failure Rates and Mixing Distributions

20.1.1 Definitions

Let $T \geq 0$ be a lifetime random variable with the cumulative distribution function (Cdf) $F(t)[\bar{F}(t) \equiv 1 - F(t)]$. Assume that $F(t)$ is indexed by a random variable Z in the following sense

$$P(T \leq t|Z = z) \equiv P(T \leq t|z) = F(t, z)$$

and that the probability density function (pdf) $f(t, z)$ exists. Then the corresponding failure rate $\lambda(t, z)$ is $f(t, z)/\bar{F}(t, z)$. Let Z be interpreted as a non-negative random variable with support $[a, b]$, $a \geq 0$, $b \leq \infty$ and pdf $\pi(z)$. Thus, a mixture Cdf is defined by

$$F_m(t) = \int_a^b F(t, z)\pi(z)\,dz\,.$$

As the failure rate is a conditional characteristic, the mixture failure rate $\lambda_m(t)$ should be defined in the following way (see, e.g., *Finkelstein* and *Esaulova* [20.9]):

$$\lambda_m(t) = \frac{\int_a^b f(t, z)\pi(z)\,dz}{\int_a^b \bar{F}(t, z)\pi(z)\,dz} = \int_a^b \lambda(t, z)\pi(z|t)\,dz\,, \tag{20.1}$$

where the conditional pdf (on the condition that $T > t$) is:

$$\pi(z|t) \equiv \pi(z|T > t) = \pi(z)\frac{\bar{F}(t, z)}{\int_a^b \bar{F}(t, z)\pi(z)\,dz}. \tag{20.2}$$

Therefore, this pdf defines a conditional random variable $Z|t$, $Z|0 \equiv Z$ with the same support. On the other hand,

consider the following *unconditional characteristic*

$$\lambda_{\mathrm{P}}(t) = \int_a^b \lambda(t, z)\pi(z)\,\mathrm{d}z\,, \tag{20.3}$$

which, in fact, defines an expected value (as a function of t) for a specific stochastic process $\lambda(t, Z)$. It follows from definitions (20.1) and (20.3) that $\lambda_{\mathrm{m}}(0) = \lambda_{\mathrm{P}}(0)$. The function $\lambda_{\mathrm{P}}(t)$ is a supplementary one, but as a trend function of a stochastic process, it captures the monotonic pattern of the family $\lambda(t, z)$. Therefore, under certain conditions, $\lambda_{\mathrm{P}}(t)$ has a similar shape to $\lambda(t, z)$: if, e.g., $\lambda(t, z)$, $z \in [a, b]$ increases with t, then $\lambda_{\mathrm{P}}(t)$ increases as well. For some specific cases (see later) $\lambda_{\mathrm{P}}(t)$ also characterizes the shape of the baseline failure rate. On the other hand, the mixture failure rate $\lambda_{\mathrm{m}}(t)$ can have a different pattern: it can ultimately decrease, for instance, or it can preserve the property that it increases with t, as in *Lynch* [20.2]. There is even the possibility of a few oscillations. However, despite all of the patterns that are possible, it will be proved that the mixture failure rate is majorized by $\lambda_{\mathrm{P}}(t)$:

$$\lambda_{\mathrm{m}}(t) < \lambda_{\mathrm{P}}(t),\ t > 0 \tag{20.4}$$

and under some additional assumptions, that

$$[\lambda_{\mathrm{P}}(t) - \lambda_{\mathrm{m}}(t)] \uparrow,\ t \geq 0. \tag{20.5}$$

Definition 20.1
[20.22]. Relation (20.4) defines a weak bending-down property for the mixture failure rate, whereas relation (20.5) is the definition of a strong bending-down property.

20.1.2 Multiplicative Model

Consider the following specific multiplicative model

$$\lambda(t, z) = z\,\lambda(t), \tag{20.6}$$

where $\lambda(t)$ is a baseline failure rate. This setting defines the widely used frailty (multiplicative) model. On the other hand, it can be also viewed as a proportional hazards (PH) model. Applying definition (20.1) gives:

$$\lambda_{\mathrm{m}}(t) = \int_a^b \lambda(t, z)\pi(z|t)\,\mathrm{d}\theta = \lambda(t)E[Z|t]. \tag{20.7}$$

The conditional expectation $E[Z|t](E[Z|0] \equiv E[Z])$ plays a crucial role in defining the shape of the mixture failure rate $\lambda_{\mathrm{m}}(t)$ in this model. The following result was proved in *Finkelstein* and *Esaulova* [20.9]:

$$E_t'[Z|t] = -\lambda(t)Var(Z|t) < 0,$$

which means that the conditional expectation of Z is a decreasing function of $t \in [0, \infty)$. On the other hand, (20.3) becomes

$$\lambda_{\mathrm{P}}(t) = \int_a^b \lambda(t, z)\pi(z)\,\mathrm{d}z = \lambda(t)E[Z|0]. \tag{20.8}$$

Therefore

$$\lambda_{\mathrm{P}}(t) - \lambda_{\mathrm{m}}(t) = \lambda(t)(E[Z|0] - E[Z|t]) > 0$$

and relation (20.4) holds, whereas under the additional sufficient condition that $\lambda(t)$ is increasing, the strong bending-down property (20.5) occurs.

20.1.3 Comparison with Unconditional Characteristics

The main additional assumption that will be needed for the following result is that the family of failure rates $\lambda(t, z)$, $z \in [a, b]$ should be ordered in z.

Theorem 20.1
Let the failure rate $\lambda(t, z)$ in the mixing model (20.1) be differentiable with respect to both arguments and be ordered as

$$\lambda(t, z_1) < \lambda(t, z_2),\ z_1 < z_2, \forall z_1, z_2 \in [a, b], t \geq 0. \tag{20.9}$$

Assume that the conditional and unconditional expectations in relations (20.1) and (20.3), respectively, are finite for $\forall t \in [0, \infty)$.

Then:

a) The mixture failure rate $\lambda_{\mathrm{m}}(t)$ bends down with time, weakly at least.

b) If, additionally, $\frac{\partial\lambda(t,z)}{\partial z}$ increases with t, then $\lambda_{\mathrm{m}}(t)$ strongly bends down with time.

Proof: It is clear that ordering (20.9) is equivalent to the condition that $\lambda(t, z)$ increases with z for each $t \geq 0$. In accordance with (20.1) and (20.3), and integrating by

parts [20.5]:

$$\begin{aligned}\Delta\lambda(t) &\equiv \int_a^b \lambda(t,z)[\pi(z)-\pi(z|t)]\,\mathrm{d}z \\ &= \lambda(t,z)[\Pi(z)-\Pi(z|t)]\big|_a^b \\ &\quad - \int_a^b \lambda'_z(t,z)[\Pi(z)-\Pi(z|t)]\,\mathrm{d}z \\ &= \int_a^b -\lambda'_z(t,z)[\Pi(z)-\Pi(z|t)]\,\mathrm{d}z > 0,\ t>0\,,\end{aligned} \tag{20.10}$$

where

$$\Pi(z) = P(Z \le z); \quad \Pi(z|t) = P(Z \le z | T > t)$$

and the term $\lambda(t,z)[\Pi(z)-\Pi(z|t)]\big|_a^b$ vanishes for $b=\infty$ as well. Inequality (20.10), and therefore the first part of the theorem, follows from $\lambda'_z(t,z)>0$ and the following inequality:

$$\Pi(z)-\Pi(z|t) < 0,\ \forall t>0, z\in(a,b). \tag{20.11}$$

Inequality (20.11) can be interpreted as: "the weakest populations die out first". This interpretation is widely used in various specific cases, especially in demographic literature [20.3]. To obtain (20.11), it is sufficient to prove that

$$\Pi(z|t) = \frac{\int_a^z \bar{F}(t,u)\pi(u)\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\pi(u)\,\mathrm{d}u}$$

increases with t, which can be easily done by considering the corresponding derivative [20.22].

The derivative $\Pi'_t(z|t) > 0$ if

$$\frac{\int_a^z \bar{F}'_t(t,u)\pi(u)\,\mathrm{d}u}{\int_a^z \bar{F}(t,u)\pi(u)\,\mathrm{d}u} > \frac{\int_a^b \bar{F}'_t(t,u)\pi(u)\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\pi(u)\,\mathrm{d}u}.$$

As $\bar{F}'_t(t,z) = -\lambda(t,z)\bar{F}(t,z)$, it is sufficient to show that

$$B(t,z) \equiv \frac{\int_a^z \lambda(t,u)\bar{F}(t,u)\pi(u)\,\mathrm{d}u}{\int_a^z \bar{F}(t,u)\pi(u)\,\mathrm{d}u}$$

increases with z. Inequality $B'_z(t,z)>0$ is equivalent to

$$\lambda(t,z)\int_a^z \bar{F}(t,u)\pi(u)\,\mathrm{d}u > \int_0^z \lambda(t,u)\bar{F}(t,u)\pi(u)\,\mathrm{d}u.$$

Thus, due to the additional assumption in Theorem 20.1b), the integrand at the end of (20.10) does increase and therefore $\Delta\lambda(t)$ does as well, which immediately leads to the strong bending-down property (20.5). ■

The following example shows the strong bending-down property of the mixture failure rate in practice.

Example 20.1: Technical devices have parameters that are also usually quite heterogeneous and should exhibit a similar deceleration in the failure rate or may even bend down practically to 0. In order to support this statement and to show that the effect of heterogeneity is significantly underestimated by most reliability practitioners, the following experiment was conducted at the Max Planck Institute for Demographic Research [20.12]. We recorded the failure times for a population of 750 miniature lamps and constructed an empirical failure rate function (in relative units) for a time interval of 250 h, which is shown in Fig. 20.1.

The results were very convincing: the failure rate initially increased (a tentative fit showed the Weibull law) and then it decreased to a very low level. This pattern for the observed failure rate is exactly the same as that predicted in *Finkelstein* and *Esaulova* [20.9] for the Weibull baseline Cdf.

We will now show now that the natural ordering for our mixing model is based on the likelihood ratio. Somewhat similar reasoning can be found in *Block*

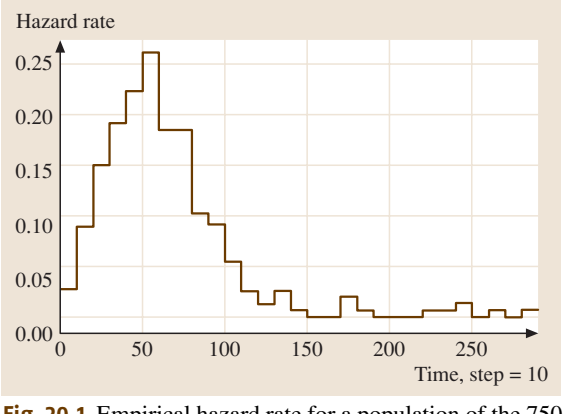


Fig. 20.1 Empirical hazard rate for a population of the 750 miniature lamps.

et al. [20.16] and *Shaked* and *Spizzichino* [20.23]. Let Z_1 and Z_2 be continuous non-negative random variables with the same support and with densities $\pi_1(z)$ and $\pi_2(z)$, respectively. Recall [20.14, 15] that Z_2 is smaller than Z_1 based on the likelihood ratio [20.24]:

$$Z_1 \geq_{\mathrm{LR}} Z_2 \,, \tag{20.12}$$

if $\pi_2(z)/\pi_1(z)$ is a decreasing function.

Definition 20.2

Let $Z(t)$, $t \in [0, \infty)$ be a family of random variables indexed by parameter t (time) with probability density functions $p(z, t)$. We say that $Z(t)$ decreases with t according to the likelihood ratio if

$$L(z, t_1, t_2) = \frac{p(z, t_2)}{p(z, t_1)}$$

decreases with z for all $t_2 > t_1$.

The following simple result states that our family of conditional mixing random variables $Z|t$, $t \in [0, \infty]$ decreases based on the likelihood ratio:

Theorem 20.2

Let the family of failure rates $\lambda(t, z)$ in the mixing model (20.1) be ordered as in the relation (20.9).

Then the family of random variables $Z|t \equiv Z|T > t$ decreases with $t \in [0, \infty)$ based on the likelihood ratio.

Proof: In accordance with (20.2):

$$L(z, t_1, t_2) = \frac{\pi(z|t_2)}{\pi(z|t_1)} = \frac{\bar{F}(t_2, z)\int_a^b \bar{F}(t_1, z)\pi(z)\,\mathrm{d}z}{\bar{F}(t_1, z)\int_a^b \bar{F}(t_2, z)\pi(z)\,\mathrm{d}z} \,. \tag{20.13}$$

Therefore, the monotonicity with z of $L(z, t_1, t_2)$ is defined by

$$\frac{\bar{F}(t_2, z)}{\bar{F}(t_1, z)} = \exp\left\{-\int_{t_1}^{t_2} \lambda(u, z)\,\mathrm{d}u\right\},$$

which, due to ordering given in (20.9), decreases with z for all $t_2 > t_1$. ■

20.1.4 Likelihood Ordering of Mixing Distributions

For the mixing model (20.1) and (20.2), consider two different mixing random variables Z_1 and Z_2 with probability density functions $\pi_1(z)$, $\pi_2(z)$ and cumulative distribution functions $\Pi_1(z)$, $\Pi_2(z)$, respectively. Assuming some type of stochastic ordering for Z_1 and Z_2, we intend to achieve simple ordering of the corresponding mixture failure rates. Using simple examples, it becomes apparent that the 'usual' stochastic ordering (stochastic dominance) is too weak to do this. It was shown in the previous section that likelihood ratio ordering is the natural one for the family of random variables $Z|t$ in our mixing model. Therefore, it seems reasonable to order Z_1 and Z_2 in this sense too.

Lemma 20.1

Let

$$\pi_2(z) = \frac{g(z)\pi_1(z)}{\int_a^b g(z)\pi_1(z)\,\mathrm{d}z} \,, \tag{20.14}$$

where $g(z)$ is a decreasing function.

Then Z_1 is stochastically larger than Z_2:

$$Z_1 \geq_{st} Z_2 \quad (\Pi_1(z) \leq \Pi_2(z),\ z \in [a, b]) \tag{20.15}$$

Proof:

$$\begin{aligned}
\Pi_2(z) &= \frac{\int_a^z g(u)\pi_1(u)\,\mathrm{d}u}{\int_a^b g(u)\pi_1(u)\,\mathrm{d}u} \\
&= \frac{\int_a^z g(u)\pi_1(u)\,\mathrm{d}u}{\int_a^z g(u)\pi_1(u)\,\mathrm{d}u + \int_z^b g(u)\pi_1(u)\,\mathrm{d}u} \\
&= \frac{g*(a, z)\int_a^z \pi_1(u)\,\mathrm{d}u}{g*(a, z)\int_a^z \pi_1(u)\,\mathrm{d}u + g*(z, b)\int_a^z \pi_1(u)\,\mathrm{d}u} \\
&\geq \int_a^z \pi_1(u)\,\mathrm{d}u = \Pi_1(z),
\end{aligned} \tag{20.16}$$

where $g*(a, z)$ and $g*(z, b)$ are the mean values of the function $g(z)$ in the corresponding integrals. As this function decreases, $g*(z, b) \leq g*(a, z)$. ■

Equation (20.14) with decreasing $g(z)$ means that $Z_1 \geq_{\mathrm{LR}} Z_2$, and it is well-known (see, e.g., [20.14]) that likelihood ratio ordering implies corresponding stochastic ordering. However, we need the previous reasoning to derive the following result.

Theorem 20.3

Let relation (20.14) (where $g(z)$ is a decreasing function) hold, which means that Z_1 is larger than Z_2 based on likelihood ratio ordering. Assume that the ordering from (20.9) holds.

Then for $\forall t \in [0, \infty)$:

$$\lambda_{\mathrm{m1}}(t) \equiv \frac{\int_a^b f(t,z)\pi_1(z)\,\mathrm{d}z}{\int_a^b \bar{F}(t,z)\pi_1(z)\,\mathrm{d}z} \geq \frac{\int_a^b f(t,z)\pi_2(z)\,\mathrm{d}z}{\int_a^b \bar{F}(t,z)\pi_2(z)\,\mathrm{d}z} \equiv \lambda_{\mathrm{m2}}(t). \tag{20.17}$$

Proof: Inequality (20.17) means that the mixture failure rate obtained for the stochastically larger (in the likelihood ratio ordering sense) mixing distribution is larger for $\forall t \in [0, \infty)$ than the one obtained for the stochastically smaller mixing distribution.

We shall first prove that

$$\Pi_1(z|t) = \frac{\int_a^z \bar{F}(t,u)\pi_1(u)\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\pi_1(u)\,\mathrm{d}u} \leq \frac{\int_a^z \bar{F}(t,u)\pi_2(u)\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\pi_2(u)\,\mathrm{d}u} \equiv \Pi_2(z|t). \tag{20.18}$$

Indeed:

$$\frac{\int_a^z \bar{F}(t,u)\pi_2(u)\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\pi_2(u)\,\mathrm{d}u} = \frac{\int_a^z \bar{F}(t,u)\frac{g(u)\pi_1(u)}{\int_a^b g(u)\pi_1(u)\,\mathrm{d}u}\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\frac{g(u)\pi_1(u)}{\int_a^b g(u)\pi_1(u)\,\mathrm{d}u}\,\mathrm{d}u} = \frac{\int_a^z g(u)\bar{F}(t,u)\pi_1(u)\,\mathrm{d}u}{\int_a^b g(u)\bar{F}(t,u)\pi_1(u)\,\mathrm{d}u} \geq \frac{\int_a^z \bar{F}(t,u)\pi_1(u)\,\mathrm{d}u}{\int_a^b \bar{F}(t,u)\pi_1(u)\,\mathrm{d}u},$$

where the last inequality follows using exactly the same argument, as in inequality (20.16) of Lemma 20.1. Similar to (20.10), and taking into account relation (20.18):

$$\begin{aligned}\lambda_{\mathrm{m1}}(t) - \lambda_{\mathrm{m2}}(t) &= \int_a^b \lambda(t,z)[\pi_1(z|t) - \pi_2(z|t)]\,\mathrm{d}z \\ &= \lambda(t,z)[\Pi_1(z|t) - \Pi_2(z|t)]\big|_a^b \\ &\quad - \int_a^b \lambda'_z(t,z)[\Pi_1(z|t) - \Pi_2(z|t)]\,\mathrm{d}z \\ &= \int_a^b -\lambda'_z(t,z)[\Pi_1(z|t) \\ &\quad - \Pi_2(z|t)]\,\mathrm{d}z \geq 0, \quad t > 0.\end{aligned} \tag{20.19}$$

The starting point of Theorem 20.3 was (20.14) with the crucial assumption of a decreasing $g(z)$ function. It should be noted, however, that this assumption can be rather formally and directly justified by considering the difference $\Delta\lambda(t) = \lambda_{\mathrm{m1}}(t) - \lambda_{\mathrm{m2}}(t)$ and using definitions (20.1) and (20.2). The corresponding numerator (the denominator is positive) is transformed into a double integral in the following way

$$\begin{aligned}&\int_a^b \lambda(t,z)\bar{F}(t,z)\pi_1(z)\,\mathrm{d}z \int_a^b \bar{F}(t,z)\pi_2(z)\,\mathrm{d}z \\ &\quad - \int_a^b \lambda(t,z)\bar{F}(t,z)\pi_2(z)\,\mathrm{d}z \int_a^b \bar{F}(t,z)\pi_1(z)\,\mathrm{d}z \\ &= \int_a^b \int_a^b \bar{F}(t,u)\bar{F}(t,s)[\lambda(t,u)\pi_1(u)\pi_2(s) \\ &\quad - \lambda(t,s)\pi_1(u)\pi_2(s)]\,\mathrm{d}u\,\mathrm{d}s \\ &= \int_{\substack{a\\u>s}}^b \int_a^b \bar{F}(t,u)\bar{F}(t,s)\{\pi_1(u)\pi_2(s)[\lambda(t,u) - \lambda(t,s)] \\ &\quad + \pi_1(s)\pi_2(u)[\lambda(t,s) - \lambda(t,u)]\}\,\mathrm{d}u\,\mathrm{d}s \\ &= \int_{\substack{a\\u>s}}^b \int_a^b \bar{F}(t,u)\bar{F}(t,s)[\lambda(t,u) - \lambda(t,s)][\pi_1(u)\pi_2(s) \\ &\quad - \pi_1(s)\pi_2(u)]\,\mathrm{d}u\,\mathrm{d}s.\end{aligned} \tag{20.20}$$

Therefore, the final double integral is positive if ordering (20.9) holds and $\pi_2(z)/\pi_1(z)$ is decreasing.

20.1.5 Ordering Variances of Mixing Distributions

Let $\Pi_1(z)$ and $\Pi_2(z)$ be two mixing distributions with equal means. It follows from equation (20.7) that, for the multiplicative model considered in this section, $\lambda_{m1}(0) = \lambda_{m2}(0)$. Intuitive considerations and reasoning based on the principle: "the weakest populations die out first" suggest that, unlike (20.17), the mixture failure rates will be ordered as $\lambda_{m1}(t) < \lambda_{m2}(t)$ for all $t > 0$ if, e.g., the variance of Z_1 is larger than the variance of Z_2. We will show that this is true for a specific case and that for a general multiplicative model this ordering only holds for a sufficiently small time t. Therefore, it is necessary to formulate a stronger condition to apply when ordering the 'variabilities' of Z_1 and Z_2.

Example 20.2: For a meaningful specific example, consider the frailty model (20.6), where Z has a gamma distribution

$$\pi(z) = \frac{\beta^\alpha}{\Gamma(\alpha)} z^{\alpha-1} \exp(-\beta z);\ \alpha > 0, \beta > 0.$$

Substituting this density into relation (20.1) gives

$$\lambda_m(t) = \frac{\lambda(t) \int_0^\infty \exp[-z\Lambda(t)] z \pi(z)\,dz}{\int_0^\infty \exp[-z\Lambda(t)]\pi(z)\,dz},$$

where $\Lambda(t) = \int_0^t \lambda(u)\,du$ is a cumulative baseline failure rate. Computing integrals results in

$$\lambda_m(t) = \frac{\alpha\lambda(t)}{\beta + \Lambda(t)}. \tag{20.21}$$

Equations (20.21) can now be written in terms of $E[Z]$ and $Var(Z)$:

$$\lambda_m(t) = \lambda(t)\frac{E^2[Z]}{E[Z] + Var(Z)\Lambda(t)}, \tag{20.22}$$

which, for the specific case $E[Z] = 1$, gives the result from *Vaupel* et al. [20.3], widely used in demography:

$$\lambda_m(t) = \frac{\lambda(t)}{1 + Var(Z)\Lambda(t)}.$$

Using equation (20.22), we can compare mixture failure rates of two populations with different Z_1 and Z_2 on the condition that $E[Z_2] = E[Z_1]$:

$$Var(Z_1) \geq Var(Z_2) \Rightarrow \lambda_{m1}(t) \leq \lambda_{m2}(t). \tag{20.23}$$

Intuitively it might be expected that this result would be valid for arbitrary mixing distributions in the multiplicative model. However, the mixture failure rate dynamics here can be much more complicated than this, even for this specific case, and this topic needs further attention in future research. A somewhat similar situation was observed in *Finkelstein* and *Esaulova* [20.9]: although the conditional variance $Var(Z|t)$ decreased with t for the multiplicative gamma frailty model, a counter example was constructed for the case of the uniform mixing distribution for [0, 1].

The following theorem shows that ordering variances is a sufficient and necessary condition for ordering mixture failure rates, but only for the initial time interval.

Theorem 20.4

Let Z_1 and Z_2 ($E[Z_2] = E[Z_1]$) be two mixing distributions in the multiplicative model (20.6), (20.7).

In this case, ordering the variances

$$Var(Z_1) > Var(Z_2) \tag{20.24}$$

is a sufficient and necessary condition for the ordering of mixture failure rates in the neighborhood of $t = 0$:

$$\lambda_{m1}(t) < \lambda_{m2}(t);\ t \in (0, \varepsilon), \tag{20.25}$$

where $\varepsilon > 0$ is sufficiently small.

Proof:

Sufficient condition:

From the results in Sect. 20.1.3:

$$\Delta\lambda(t) = \lambda_{m1}(t) - \lambda_{m2}(t) = \lambda(t)(E[Z_1|t] - E[Z_2|t]), \tag{20.26}$$

$$E_t'[Z_i|t] = -\lambda(t)Var(Z_i|t) < 0,\ i = 1, 2, t \geq 0, \tag{20.27}$$

where

$$E[Z_i|0] \equiv E[Z_i],\ Var(Z_i|t) \equiv Var(Z_i). \tag{20.28}$$

As the means of the mixing variables are equal, relation (20.26) for $t = 0$ reads $\Delta\lambda(0) = 0$, and therefore the time interval in (20.25) is opened. Thus, if the ordering in (20.24) holds, the ordering in (20.25) then follows immediately after, considering the derivative of

$$\frac{\lambda_{m1}(t)}{\lambda_{m2}(t)} = \frac{E[Z_1|t]}{E[Z_2|t]}$$

at $t = 0$ and taking into account relations (20.27) and notation (20.28).

Necessary condition:

Similar to (20.20), the numerator of the difference $\Delta\lambda(t)$ is

$$\lambda(t)\int_a^b\int_a^b\{\exp[-\Lambda(t)(u+s)]\}(u-s)\pi_1(u)\pi_2(s)\,\mathrm{d}u\,\mathrm{d}s,$$

where, as previously, $\Lambda(t)=\int_0^t\lambda(u)\,\mathrm{d}u$. After changing variables to $x=(u+s)/2$, $y=(u-s)/2$, the double integral is transformed to the iterated integral and denoted by $G(t)$:

$$G(t)\equiv\int_a^b\exp[-2\Lambda(t)x]\times\int_{-x}^{x}y\pi_1(x+y)\pi_2(x-y)\,\mathrm{d}y\,\mathrm{d}x. \qquad (20.29)$$

Denote the internal integral in (20.29) by $g(x)$. Then:

$$G(t)=\int_a^b\{\exp[-2\Lambda(t)x]\}g(x)\,\mathrm{d}x.$$

On the other hand, reverting back to the initial variables of integration and taking into account that $\Lambda(0)=0$, we get

$$\begin{aligned}G(0)&=\int_a^b g(x)\,\mathrm{d}x=\int_a^b\int_a^b(u-s)\pi_1(u)\pi_2(s)\,\mathrm{d}u\,\mathrm{d}s\\&=\int_a^b u\pi_1(u)\,\mathrm{d}u-\int_a^b u\pi_2(u)\,\mathrm{d}u\\&=E[Z_1]-E[Z_2]=0.\end{aligned}$$

Assume, firstly, that $\lambda(0)\neq 0$. As $G(0)=0$, the function $G(t)$ is negative in the neighborhood of 0 if $G'(0)<0$:

$$G'(t)=-2\lambda(t)\int_a^b\{\exp[-2\Lambda(t)x]\}xg(x)\,\mathrm{d}x,$$

$$G'(0)<0\Rightarrow\int_a^b xg(x)\,\mathrm{d}x>0.$$

If $\Delta\lambda(t)<0$, $t\in(0,\varepsilon)$ [condition (20.25)], then $G(t)<0$, $t\in(0,\varepsilon)$, and taking into account that

$$\begin{aligned}\int_a^b xg(x)\,\mathrm{d}x&=\int_a^b\int_a^b\frac{u+s}{2}(u-s)\pi_1(s)\pi_2(s)\,\mathrm{d}u\,\mathrm{d}s\\&=\frac{1}{2}\int_a^b\int_a^b(u^2-s^2)\pi_1(u)\pi_2(s)\,\mathrm{d}u\,\mathrm{d}s\\&=\frac{1}{2}[Var(Z_1)-Var(Z_2)],\end{aligned}$$

we arrive at the ordering given in (20.24).

Similar considerations are valid for $\lambda(0)=0$. The function $G(t)$ is negative in this case in the neighborhood of 0 if $G''(0)<0$. As

$$G''(0)=-2\lambda'(0)\int_a^b xg(x)\,\mathrm{d}x$$

and $\lambda'(0)>0$ [since $\lambda(t)>0$, $t>0$ and $\lambda(0)=0$], the same reasoning used for the case $\lambda(0)\neq 0$, also holds here. ■

A trivial but important consequence of this theorem is as follows:

Corollary

Let mixture failure rate ordering (20.25) hold for $t\in(0,\infty)$. Then inequality (20.24) holds.

20.2 Modeling the Impact of the Environment

20.2.1 Bounds in the Proportional Hazards Model

Consider the specific multiplicative frailty model (20.6) and (20.7). Formally combine this model with the proportional hazards (PH) model in a following way:

$$\lambda(t,z,k)=zk\lambda(t)\equiv z_k\lambda(t). \qquad (20.30)$$

Therefore, the baseline $F(t)$ is indexed by the random variable $Z_k=kZ$ with the pdf $\pi_k(z)=\pi(z/k)$, whereas the corresponding conditional pdf $\pi_k(z|t)$ is given by the right hand side of (20.2), where $\pi(z)$ is substituted by $\pi_k(z)$. Equivalently, (20.30) can be interpreted as a frailty model with a mixing random variable Z and a baseline failure rate $k\lambda(t)$. These two simple and equivalent interpretations will help us in what follows. Without losing

any generality, assume that $a = 0$ and $b = \infty$. Thus, similar to (20.6)–(20.9), the mixture failure rate in this case is

$$\lambda_{mk}(t) = k\lambda(t) \int_0^\infty z\pi_k(z|t)\,dz \equiv \lambda(t)E[Z_k|t]. \quad (20.31)$$

As $Z_k = kZ$, its density function is

$$\pi_k(z) = \frac{1}{k}\pi\left(\frac{z}{k}\right).$$

Theorem 20.5

Let the mixture failure rates for the multiplicative models (20.6) and (20.30) be given by relations (20.7) and (20.31), respectively, where $k > 1$.

Assume that the following quotient increases with z:

$$\frac{\pi_k(z)}{\pi(z)} = \frac{\pi\left(\frac{z}{k}\right)}{k\pi(z)} \uparrow \quad (20.32)$$

Then:

$$\lambda_{mk}(t) > \lambda_m(t);\ \forall t \in [0, \infty). \quad (20.33)$$

Proof: Although inequality (20.33) seems rather trivial at first sight, it is only valid for some specific cases of mixing (e.g., the multiplicative model). It is clear that (20.33) is always true for sufficiently small t, whereas with larger t the ordering can be different for general mixing models. Denote:

$$\Delta\lambda_m(t) = \lambda_{mk}(t) - \lambda_m(t).$$

Using definitions (20.1)–(20.2), it can be seen that, similar to the case for relation (20.20), the sign of this difference is defined by the sign of

$$\begin{aligned}
&\int_0^\infty z\bar{F}(t,z)\pi_k(z)\,dz \int_0^\infty \bar{F}(t,z)\pi(z)\,dz \\
&- \int_0^\infty z\bar{F}(t,z)\pi_k(z)\,dz \int_0^\infty \bar{F}(t,z)\pi(z)\,dz \\
&= \int_0^\infty \int_0^\infty \bar{F}(t,u)\bar{F}(t,s)[u\pi_k(u)\pi(s) \\
&\quad - s\pi_k(u)\pi(s)]\,du\,ds \\
&= \int_{\substack{0\\u>s}}^\infty \int_0^\infty \bar{F}(t,u)\bar{F}(t,s)[\pi_k(u)\pi(s)(u-s) \\
&\quad + \pi_k(s)\pi(u)(s-u)]\,du\,ds \\
&= \int_{\substack{0\\u>s}}^\infty \int_0^\infty \bar{F}(t,u)\bar{F}(t,s)(u-s)[\pi_k(u)\pi(s) \\
&\quad - \pi_k(s)\pi(u)]\,du\,ds. \quad (20.34)
\end{aligned}$$

Therefore, the sufficient condition for inequality (20.33) is condition (20.32), which is, in fact, rather crude. It is easy to verify that this condition is satisfied, for example, for the gamma and the Weibull densities, which are often used for mixing. ■

Example 20.3: Consider the same setting as in Example 20.1. Condition (20.32) is satisfied for the gamma distribution. The mixture failure rate $\lambda_m(t)$ in this case is given by relation (20.22). A similar equation obviously exists for $\lambda_{mk}(t)$, and the corresponding comparison can be performed explicitly:

$$\begin{aligned}
\lambda_{mk}(t) &= \lambda(t)\frac{E^2[Z_k]}{E[Z_k] + Var(Z_k)\Lambda(t)} \\
&= \lambda(t)\frac{k^2E^2[Z]}{kE[Z] + k^2Var(Z)\Lambda(t)} > \lambda_m(t).
\end{aligned} \quad (20.35)$$

Now we shall obtain an upper bound for $\lambda_{mk}(t)$.

Theorem 20.6

Let the mixture failure rates for the multiplicative models (20.6) and (20.30) be given by relations (20.7) and (20.31), respectively, where $k > 1$.

Then:

$$\lambda_{mk}(t) < k\lambda_m(t);\ \forall t \in (0, \infty). \quad (20.36)$$

Proof: Consider the difference $\lambda_{mk}(t) - k\lambda_m(t)$ similarly to (20.34), but in a slightly different way: $\lambda_{mk}(t)$ will be equivalently defined by the baseline failure rate $k\lambda(t)$ and the mixing variable Z (in (20.34) it was defined by the baseline $\lambda(t)$ and the mixing variable kZ). This means that:

$$\lambda_{mk}(t) - k\lambda_m(t) = k\lambda(t)(\hat{E}[Z|t] - E[Z|t]), \quad (20.37)$$

where the conditioning in $\hat{E}[Z|t]$ is different from the one in $E[Z|t]$ in the sense described. Denote:

$$\bar{F}_k(t,z) = \exp\left[-zk\Lambda(t)\right].$$

'Symmetrically' to (20.34), $\text{sign}[\lambda_{mk}(t) - k\lambda_m(t)]$ is defined by

$$\text{sign} \int_{\substack{0 \\ u>s}}^{\infty} \int_{0}^{\infty} \pi(u)\pi(s)(u-s)[\bar{F}_k(t,u)\bar{F}(t,s) - \bar{F}(t,u)\bar{F}_k(t,s)]\,du\,ds,$$

which is negative for all $t > 0$ since

$$\frac{\bar{F}_k(t,z)}{\bar{F}(t,z)} = \exp[-(k-1)z\Lambda(t)]$$

decreases with z. ■

It is worth noting that we do not need an additional condition for this bound as in the case of Theorem 20.5. Also, it is clear that $\lambda_{mk}(0) = k\lambda_m(0)$. As previously mentioned, model (20.30) defines a combination of a PH model and a frailty model. When $Z = 1$, it is an 'ordinary' PH model. In the presence of a random Z, as follows from (20.36), the observed failure rate $\lambda_{mk}(t)$ cannot be obtained as $k\lambda_m(t)$ due to the nature of the mixing.

Therefore, the PH model in each realization does not result in the PH model for the corresponding mixture failure rate.

Example 20.3 (continuation): We illustrate inequality (20.36):

$$\lambda_{mk}(t) = \lambda(t)\frac{k^2E^2[Z]}{kE[Z]+k^2Var(Z)\Lambda(t)} < \lambda(t)\frac{kE^2[Z]}{E[Z]+Var(Z)\Lambda(t)} = k\lambda_m(t).$$

20.2.2 Change Point in the Environment

Assume that there are two possible environments (stresses), $\varepsilon(t)$ and $\varepsilon_s(t)$: the baseline and a more severe one, respectively. The baseline environment for our heterogeneous population corresponds to the observed failure rate $\lambda_m(t)$ and the more severe one to $\lambda_{mk}(t)$, $k > 1$. As we did previously, assume also that the PH model for each subpopulation (for each fixed z) holds. Consider a piece-wise constant step stress with a single change point at t_1:

$$\varepsilon(t_1) = \begin{cases} \varepsilon, & 0 \le t < t_1, \\ \varepsilon_k & t \ge t_1, \end{cases} \tag{20.38}$$

where the stresses ε and ε_k correspond to the failure rates $z\lambda(t)$ and $zk\lambda(t)$, respectively ($k > 1, z \ge 0$). In accordance with the 'memoryless property' of the PH model, the stress (20.38) results in the following failure rate for each subpopulation:

$$\lambda(t, t_1, z, k) = \begin{cases} z\lambda(t), & 0 \le t < t_1 \\ kz\lambda(t) & t \ge t_1 \end{cases} \tag{20.39}$$

Denote the resulting mixture failure rate in this case as:

$$\lambda_m(t, t_1) = \begin{cases} \lambda_m(t), & 0 \le t < t_1, \\ \tilde{\lambda}_{mk}(t) & t \ge t_1, \end{cases} \tag{20.40}$$

where, similar to the previous section,

$$\tilde{\lambda}_{mk}(t_1) = k\lambda_m(t_1). \tag{20.41}$$

It is worth noting that relation (20.41) means that this model with a step stress is proportional for the mixture failure rates only at the switching point t_1. We want to prove the following inequality:

$$\lambda_{mk}(t) < \tilde{\lambda}_{mk}(t);\ \forall t \in [t_1, \infty). \tag{20.42}$$

In accordance with (20.40), consider two **initial** (for the interval $[0, \infty)$) mixing distributions: $Z_1 = Z|T_1 > t_1$, where T_1 is defined by the baseline failure rate $k\lambda(t)$ and $\tilde{Z}_1 = Z|\tilde{T}_1 > t_1$, where $\tilde{T}_1$ is defined by the baseline failure rate $\lambda(t)$. As follows from definition (20.2), the corresponding ratio

$$\frac{\tilde{\pi}(z,t_1)}{\pi(z,t_1)} = \exp[(k-1)z\Lambda(t_1)]$$

increases with z. Then inequality (20.42) follows immediately after taking the proof of Theorem 20.1 into account with obvious alterations caused by the change in the left end point of the interval from 0 to t_1.

Inequality (20.42) was graphically illustrated in *Vaupel* and *Yashin* ([20.25] Fig.10) for a specific case of a discrete mixture of two subpopulations and the Gompertz baseline failure rate. The demographic meaning of this was the following: suppose we decrease the mortality rates of the subpopulations during early life ($[0, t_1)$). Then the observed mortality rate for $[t_1, \infty)$ is *larger* than the observed mortality rate for the initial mixture without changes. In other words, early success results in more failure later on [20.25].

20.2.3 Shocks in Heterogeneous Populations

Now consider the general mixing model (20.1)–(20.2) and assume that an instantaneous shock occurs at time $t = t_1$. This shock affects the whole population: with corresponding complementary probabilities it either kills an individual or 'leaves him unchanged'. Without losing any generality, let $t_1 = 0$; otherwise a new initial mixing variable $Z|t_1$ needs to be defined and the corresponding procedure can be easily adjusted to this case. It is natural to suppose that the frailest individuals or populations (those with the largest failure rates) are more susceptible to being killed.

This setting can be defined probabilistically in the following way. Let $\pi_1(z)$ denote the frailty distribution of a random variable Z_1 after the shock and let $\lambda_{ms}(t)$ be the corresponding observed (mixture) failure rate after the shock. Assume that

$$\pi_1(z) = \frac{g(z)\pi(z)}{\int_a^b g(z)\pi(z)\,dz}, \tag{20.43}$$

where $g(z)$ is a decreasing function and therefore $\pi_1(z)/\pi(z)$ also decreases. This means that the shock performs a kind of burn-in operation [20.16] and the random variables Z and Z_1 are ordered based on the likelihood ratio [20.14, 15]:

$$Z \geq_{LR} Z_1 \tag{20.44}$$

We are now able to formulate the following result.

Theorem 20.7

Let relation (20.43), which defines the mixing density after a shock at $t = 0$ (and where $g(z)$ is a decreasing function), hold.

Also assume that the ordering given by (20.9) holds Then:

$$\lambda_{ms}(t) < \lambda_m(t);\ \forall t \in [0, \infty). \tag{20.45}$$

Proof: Inequality (20.9) is a natural ordering of the family of failure rates $\lambda(t, z)$, $z \in [0, \infty)$ and it trivially holds for the specific model (20.6). Performing all of the steps we used when obtaining relation (20.34), we finally obtain:

$$\begin{aligned}&\operatorname{sign}[\lambda_{ms}(t) - \lambda_m(t)]\\ &= sign \int_{\substack{a\\ u>s}}^{b} \int_a^b \bar{F}(t,u)\bar{F}(t,s)[\lambda(t,u)\\ &- \lambda(t,s)][\pi_1(u)\pi(s) - \pi_1(s)\pi(u)]\,du\,ds,\end{aligned}$$

which is negative due to definition (20.43) and the assumptions of this theorem. ■

At $t = 0$, for instance:

$$\lambda_m(0) - \lambda_{ms}(0) = \int_0^\infty \lambda(0,z)[\pi(z) - \pi_1(z)]\,dz.$$

In accordance with inequality (20.45), the curve $\lambda_{ms}(t)$ lies beneath the curve $\lambda_m(t)$ for $t \geq 0$. This fact seems intuitively evident, but, in fact, it is only valid due to the rather stringent conditions of this theorem. It can be shown, for instance, that replacing condition (20.44) with a weaker one of stochastic dominance, $Z_{st} \geq Z_1$, will not guarantee the ordering given in (20.45) for all t.

This result can be generalized to a sequence of shocks of the type described that occur at times $\{t_i\}$, $i = 1, 2, \ldots$

20.3 Asymptotic Behaviors of Mixture Failure Rates

20.3.1 Survival Model

The asymptotic behaviors of mixture failure rates have been studied by *Block* et al. [20.16], *Gurland* and *Sethuraman* [20.20], *Lynn* and *Singpurwalla* [20.19] and *Block* et al. [20.17], to name but a few. In *Finkelstein* and *Esaulova* [20.9] we considered the properties of $\lambda_m(t)$ as $t \to \infty$ for the multiplicative model (20.6). As $\lambda_m(t) = \lambda(t)E[Z|t]$, this product was analyzed for increasing $\lambda(t)$ and conditions implying convergence to 0 were derived, taking into account that the conditional expectation $E[Z|t]$, defined in (20.7), decreases with t. The approach taken in this section is different: we study a new lifetime model and derive explicit asymptotic formulae for mixture failure rates that generalize various specific results obtained for proportional hazards and additive hazards models. This approach also allows us to deal with the accelerated life model (ALM), which has not been studied in the literature.

We now define a class of distributions $F(t, z)$ and study the asymptotic behavior of the corresponding mixture failure rate $\lambda_m(t)$. To begin with it is more convenient to define this in terms of the cumulative failure rate $\Lambda(t, z)$, rather than in terms of $\lambda(t, z)$:

$$\Lambda(t, z) = A[z\phi(t)] + \psi(t). \tag{20.46}$$

General Assumptions for the Model (20.46):
The natural properties of the cumulative failure rate of the absolutely continuous distribution $F(t, z)$ (for $\forall z \in [0, \infty)$) imply that the functions $A(s)$, $\phi(t)$ and $\psi(t)$ are differentiable, the right hand side of (20.46) does not decrease with t and it tends to infinity as $t \to \infty$ and $A[z\phi(0)] + \psi(0) = 0$. Therefore, these properties will be assumed throughout this section, although some of them will not be needed for formal proofs.

An important additional simplifying assumption is that

$$A(s), s \in [0, \infty);\ \phi(t), t \in [0, \infty) \tag{20.47}$$

are increasing functions of their arguments and $A(0) = 0$, although some generalizations (e.g., only for ultimately increasing functions) are easily performed. Therefore, we will view $1 - \exp[-A(z\phi(t))]$, $z \neq 0$ in this chapter as a lifetime Cdf.

It should be noted that model (20.46) can be also easily generalized to the form $\Lambda(t, z) = A[g(z)\phi(t)] + \psi(t) + \eta(z)$ for some properly defined functions $g(z)$ and $\eta(z)$. However, we cannot generalize any further (at least, at this stage), and the multiplicative form of the arguments in $A[g(z)\phi(t)]$ is important to our method of deriving asymptotic relations. It is also clear that the additive term $\psi(t)$, although important in applications, provides only a slight generalization for further analysis of $\lambda_m(t)$, as (20.46) can be interpreted in terms of two components in series (or, equivalently, as two competing risks).

The failure rate, which corresponds to the cumulative failure rate $\Lambda(t, z)$, is

$$\lambda(t, z) = z\phi'(t)A'[z\phi(t)] + \psi'(t), \tag{20.48}$$

where, by $A'[z\phi(t)]$, we in fact mean $\mathrm{d}A[z\phi(t)]/\mathrm{d}[z\phi(t)]$.

Now we can explain why we start with the cumulative failure rate and not with the failure rate itself, which is common in lifetime modeling. The reason is that one can easily suggest intuitive interpretations of (20.46), whereas it is certainly not as simple to interpret the failure rate structure in the form (20.48) without stating that it just follows from the structure of the cumulative failure rate.

Relation (20.46) defines a rather broad class of survival models which can be used, for example, to model the impact of the environment on survival characteristics. The proportional hazards, additive hazards and accelerated life models, widely used in reliability, survival analysis and risk analysis, are the obvious specific cases of our relations (20.46) or (20.48):

PH (multiplicative) model:
Let

$$A(u) \equiv u,\ \phi(t) = \Lambda(t),\ \psi(t) \equiv 0.$$

Then

$$\lambda(t, z) = z\lambda(t), \quad \Lambda(t, z) = z\Lambda(t). \tag{20.49}$$

ALM:
Let

$$A(u) \equiv \Lambda(u),\ \phi(t) = t,\ \psi(t) \equiv 0.$$

Then

$$\Lambda(t, z) = \int_0^{tz} \lambda(u)\,\mathrm{d}u, \quad \lambda(t, z) = z\lambda(tz). \tag{20.50}$$

AH *model*:
Let

$$A(u) \equiv u,\ \phi(t) = t,\ \psi(t) \text{ is increasing}, \psi(0) = 0.$$

Then

$$\lambda(t, z) = z + \psi'(t), \quad \Lambda(t, z) = zt + \psi(t). \tag{20.51}$$

The functions $\lambda(t)$ and $\psi'(t)$ act as baseline failure rates in equations (20.49), (20.50) and (20.51), respectively. Note that, in all of these models, the functions $\phi(t)$ and $A(s)$ increase monotonically.

The asymptotic behaviors of the mixture failure rates for the PH and AH models have been studied for some specific mixing distributions in, for example, *Gurland* and *Sethuraman* [20.20] and *Finkelstein* and *Esaulova* [20.9]. On the other hand, as far as we know, the mixture failure rate for the ALM has only been considered at a descriptive level in *Anderson* and *Louis* [20.26].

20.3.2 Main Result

The next theorem derives an asymptotic formula for the mixture failure rate $\lambda_m(t)$ under rather mild assumptions. We use an approach related to the ideology of generalized convolutions, for example *Laplace* and *Fourier* transforms and (especially) Mellin convolutions [20.27].

Theorem 20.8

Let the cumulative failure rate $\Lambda(t, z)$ be given by the model (20.46) and the mixing pdf $\pi(z)$ be defined as

$$\pi(z) = z^{\alpha}\pi_1(z), \tag{20.52}$$

where $\alpha > -1$ and $\pi_1(z)$, $\pi_1(0) \neq 0$ is a function that is bounded in $[0, \infty)$ and continuous at $z = 0$.

Assume also that $\phi(t)$ increases to infinity:

$$\phi(t) \to \infty \quad \text{as} \quad t \to \infty \tag{20.53}$$

and that

$$\exp[-A(s)]s^{\alpha+1} \to 0 \quad \text{as} \quad t \to \infty,$$
$$\int_0^{\infty} \exp[-A(s)]s^{\alpha}\,\mathrm{d}s < \infty. \tag{20.54}$$

Then

$$\lambda_{\mathrm{m}}(t) - \psi'(t) \sim (\alpha+1)\frac{\phi'(t)}{\phi(t)}. \tag{20.55}$$

By relation (20.55) we (as usual) mean asymptotic equivalence, and we write $a(t) \sim b(t)$ as $t \to \infty$, if $\lim_{t\to\infty}[a(t)/b(t)] = 1$.

Proof: Firstly, we need a simple lemma for the Dirac sequence of functions. ■

Lemma 20.2

Let $g(z)$, $h(z)$ be non-negative functions in $[0, \infty)$ that satisfy the following conditions:

$$\int_0^{\infty} g(z)\,\mathrm{d}z < \infty, \tag{20.56}$$

and $h(z)$ is bounded and continuous at $z = 0$.

Then, as $t \to \infty$:

$$t\int_0^{\infty} g(tz)h(z)\,\mathrm{d}z \to h(0)\int_0^{\infty} g(z)\,\mathrm{d}z. \tag{20.57}$$

Proof: Substituting $u = tz$,

$$t\int_0^{\infty} g(tz)h(z)\,\mathrm{d}z = \int_0^{\infty} g(u)h(u/t)\,\mathrm{d}u.$$

The function $h(u)$ is bounded and $h(u/t) \to 0$ as $t \to \infty$, thus convergence (20.57) holds by the dominated convergence theorem. ■

We are now able to prove Theorem 20.8. The proof is straightforward, as we use definition (20.1) and Lemma 20.2.

The survival function for the model (20.46) is

$$\bar{F}(t, z) = \exp\{-[A(z\phi(t)] - \psi(t)\}.$$

Taking into account that $\phi(t) \to \infty$ as $t \to \infty$, and applying Lemma 20.2 to the function $g(u) = \exp[-A(u)]u^{\alpha}$:

$$\int_0^{\infty} \bar{F}(t, z)\pi(z)\,\mathrm{d}z = \int_0^{\infty} \exp\{-[A(z\phi(t))] - \psi(t)\}z^{\alpha}\pi_1(z)\,\mathrm{d}z$$
$$\sim \frac{\exp[-\psi(t)]\pi_1(0)}{\phi(t)^{\alpha+1}}\int_0^{\infty} \exp[-A(s)]s^{\alpha}\,\mathrm{d}s, \tag{20.58}$$

where the integral is finite due to the condition given in (20.54).

The corresponding probability density function is:

$$\begin{aligned} f(t, z) =& \{A'[z\phi(t)]z\phi'(t) \\ &+ \psi'(t)\}\exp\{-A[z\phi(t)] - \psi(t)\} \\ =& A'[z\phi(t)]z\phi'(t)\exp\{-A[z\phi(t)] \\ &- \psi(t)\} + \psi'(t)\bar{F}(t, z). \end{aligned}$$

Similarly, applying Lemma 20.2 gives:

$$\int_0^{\infty} f(t, z)\pi(z)\,\mathrm{d}z - \psi'(t)\int_0^{\infty} \bar{F}(t, z)\pi(z)\,\mathrm{d}z$$
$$= \phi'(t)\exp[-\psi(t)]\int_0^{\infty} A'[z\phi(t)]$$
$$\exp\{-A[z\phi(t)]\}z^{\alpha+1}\pi_1(z)\,\mathrm{d}z$$
$$\sim \frac{\phi'(t)\exp[-\psi(t)]\pi_1(0)}{\phi(t)^{\alpha+2}}\int_0^{\infty} A'(s)$$
$$\exp[-A(s)]s^{\alpha+1}\,\mathrm{d}s \tag{20.59}$$

Integrating by parts and using condition (20.54):

$$\int_0^{\infty} A'(s)\exp[-A(s)]s^{\alpha+1}\,\mathrm{d}s$$
$$= (\alpha+1)\int_0^{\infty} \exp[-A(s)]s^{\alpha}\,\mathrm{d}s. \tag{20.60}$$

Combining relations (20.58)–(20.60), finally:

$$\frac{\int_0^\infty f(t,z)\pi(z)\,dz}{\int_0^\infty \bar{F}(t,z)\pi(z)\,dz} - \psi'(t) \sim (\alpha+1)\frac{\phi'(t)}{\phi(t)}.$$

It is easy to see that assumption (20.52) holds for the main lifetime distributions, such as Weibull, gamma, log-normal etc. Assumption (20.53) states a natural condition for the function $\phi(t)$, which can often be viewed as a scale transformation. Conditions (20.54) mean that the Cdf $1-\exp[-A(s)]$ should not be 'too heavy-tailed' (as e.g. the Pareto distribution $1-s^{-\beta}$, for $s \geq 1, \beta-\alpha > 1$) and are equivalent to the condition that a moment of order $\alpha+1$ exists for this Cdf. The examples shown in the next subsection will clearly illustrate that these conditions are not stringent at all and can be easily met in most practical situations.

A crucial feature of this result is that the asymptotic behavior of the mixture failure rate depends only [omitting an obvious additive term $\psi(t)$] on the behavior of the mixing distribution near to zero and on the derivative of the logarithm of the scale function $\phi(t)$: $[\log \phi(t)]' = \phi'(t)/\phi(t)$. When $\pi(0) \neq 0$ and $\pi(z)$ is bounded in $[0, \infty)$, the result does not depend on the mixing distribution at all, as $\alpha = 0$!

20.3.3 Specific Models

Multiplicative (PH) Model

In the conventional notation, the baseline failure rate is usually denoted by $\lambda_0(t)$ [or $\lambda_b(t)$]. Therefore, model (20.6) reads:

$$\lambda(t,z) = z\lambda_0(t), \quad \Lambda_0(t) = \int_0^t \lambda_0(u)\,du \tag{20.61}$$

and the mixture failure rate is given by

$$\lambda_m(t) = \frac{\int_0^\infty z\lambda_0(t)\exp[-z\Lambda_0(t)]\pi(z)\,dz}{\int_0^\infty \exp[-z\Lambda_0(t)]\pi(z)\,dz}. \tag{20.62}$$

As $A(u) \equiv u, \phi(t) = \Lambda_0(t), \psi(t) \equiv 0$ in this specific case, Theorem 20.8 simplifies to:

Corollary 20.1

Assume that the mixing pdf $\pi(z), z \in [0, \infty)$ can be written as

$$\pi(z) = z^\alpha \pi_1(z), \tag{20.63}$$

where $\alpha \geq -1$ and $\pi_1(z)$ is bounded in $[0, \infty)$, continuous at $z = 0$ and $\pi_1(0) \neq 0$.

Then the mixture failure rate for the multiplicative model (20.61) has the following asymptotic behavior:

$$\lambda_m(t) \sim \frac{(\alpha+1)\lambda_0(t)}{\int_0^t \lambda_0(u)\,du}. \tag{20.64}$$

The mixture failure rate given by (20.62) can be obtained explicitly when the Laplace transform of the mixing pdf $\tilde{\pi}(t)$ is easily computed. As the cumulative failure rate increases monotonically with t, the mixture survival function is written in terms of the Laplace transform as:

$$\int_0^\infty \exp[-z\Lambda_0(t)]\pi(z)dz = \tilde{\pi}[\Lambda_0(t)].$$

Therefore, (20.62) becomes:

$$\lambda_m(t) = -\frac{(\tilde{\pi}[\Lambda_0(t)])'}{\tilde{\pi}[\Lambda_0(t)]} = -(\log \tilde{\pi}[\Lambda_0(t)])'$$

and the corresponding inverse problem can also be solved; in other words, given the mixture failure rate and the mixing distribution, obtain the baseline failure rate [20.28].

Example 20.4: As for examples 20.1 and 20.3, consider a frailty model (20.61) where Z has a gamma distribution, which, for notational convenience, is written in a slightly different form:

$$\pi(z) = \left(\frac{z}{b}\right)^{c-1} \exp\left\{-\frac{z}{b}\right\} \frac{1}{b\Gamma(c)}, \tag{20.65}$$

where $b, c > 0$.

The expected value Z is bc and the variance is b^2c. The Laplace transform of $\pi(z)$ is $\tilde{\pi}(t) = c(tb+1)^{-c}$ and therefore the mixture failure rate is given by the following expression, which is the same as (20.22):

$$\lambda_m(t) = \frac{bc\lambda_0(t)}{1 + b\int_0^t \lambda_0(u)\,du}. \tag{20.66}$$

Obviously, the asymptotic behavior of $\lambda_m(t)$ can be analyzed explicitly. Consider two specific cases.

If the baseline distribution is Weibull with $\lambda_0(t) = \lambda t^\beta$, $\beta \geq 0$, then the mixture failure rate (20.66) is

$$\lambda_m(t) = \frac{(\beta+1)\lambda bct^\beta}{(\beta+1)+\lambda bt^{\beta+1}}, \tag{20.67}$$

which converges to 0 as $t \to \infty$ because it is $\sim(\beta+1)ct^{-1}$, exactly as prescribed by formula (20.64) of Corollary 20.1 ($c = \alpha+1$).

If the baseline distribution is Gompertz with $\lambda_0(t) = \mu\exp(\beta t)$, then a simple transformations gives

$$\lambda_m(t) = \frac{\beta c\exp(\beta t)}{\exp(\beta t)+\left(\frac{\beta}{\mu b}-1\right)}. \tag{20.68}$$

If $b = \beta/\mu$, then $\lambda_m(t) \equiv \beta c$; if $b > \beta/\mu$, then $\lambda_m(t)$ increases to β/μ; if $b < \beta/\mu$, it decreases to β/μ.

It is reasonable to compare the asymptotic behaviors of (20.67) and (20.68) for the same mixing distribution (20.65). For the Weibull Cdf, the mixture failure rate converges to 0. This means that, within the framework of the multiplicative model, where the family of failure rates is ordered in z, we can still speak of convergence to the failure rate of the strongest population, defining the $z = 0$ case as a 'generalized' (or formal) strongest failure rate: $\lambda(t, 0) = 0$. However, the failure rate for a Gompertz Cdf does not converge to 0 – it converges to a constant, thus violating the principle of converging to the failure rate of the strongest population, even when formulated in a 'generalized' form! The reason for this is the sharp increase in the function $\phi(t)$, which is proportional to $\exp(\beta t)$ in the latter case.

Accelerated Life Model

In the conventional notation, this model is written as:

$$\lambda(t, z) = z\lambda_0(tz),$$
$$\Lambda_0(tz) = \int_0^{tz} \lambda_0(u)\,du. \tag{20.69}$$

Although the ALM also has a very simple definition, the presence of the mixing parameter z in the arguments make analysis of the mixture failure rate more complex than in the multiplicative case. Therefore, as mentioned previously, this model is practically unstudied. The mixture failure rate in this specific case is

$$\lambda_m(t) = \frac{\int_0^\infty z\lambda_0(tz)\exp[-\Lambda_0(tz)]\pi(z)\,dz}{\int_0^\infty \exp[-\Lambda_0(tz)]\pi(z)\,dz}. \tag{20.70}$$

The asymptotic behavior of $\lambda_m(t)$ can be described as a specific case of Theorem 20.8 with $A(s) = \Lambda_0(s)$, $\phi(t) = t$ and $\psi(t) \equiv 0$:

Corollary 20.2

Assume that the mixing pdf $\pi(z)$, $z \in [0, \infty)$ can be defined as $\pi(z) = z^\alpha\pi_1(z)$, where $\alpha > -1$, $\pi_1(z)$ is continuous at $z = 0$ and bounded in $[0, \infty)$, $\pi_1(0) \neq 0$.

Let the baseline distribution with cumulative rate $\Lambda_0(t)$ have a moment of order $\alpha+1$. Then

$$\lambda_m(t) \sim \frac{\alpha+1}{t} \tag{20.71}$$

as $t \to \infty$.

The conditions of Corollary 20.2 are not that strong and are relatively natural. Most widely used lifetime distributions have all of the moments. The Pareto distribution will be discussed in the next example.

As already stated, the conditions on the mixing distribution hold for the gamma and the Weibull distributions, which are commonly used as mixing distributions.

Relation (20.71) is really surprising, as it does not depend on the baseline distribution, which seems strange, at least at first sight. It is also dramatically different to the multiplicative case (20.64). It follows from Example 20.4 that both asymptotic results coincide in the case of the Weibull baseline distribution; this is obvious, as the ALM can only be reparameterized to end up with a PH model and vice versa for the Weibull distribution.

The following example shows other possible asymptotic behaviors for $\lambda_m(t)$ when one of the conditions of Corollary 20.2 does not hold.

Example 20.5: Consider the gamma mixing distribution $\pi(z) = z^\alpha\exp(-x)/\Gamma(\alpha+1)$. Let the baseline distribution be the Pareto distribution with density $f_0(t) = \beta/t^{\beta+1}$ $t \geq 1$, $\beta > 0$.

For $\beta > \alpha+1$ the conditions of Corollary 20.2 hold and relation (20.71) occurs. Let $\beta \leq \alpha+1$, which means that the baseline distribution doesn't have the $(\alpha+1)$th moment. Therefore, one of the conditions of Corol-

lary 20.2 does not hold. In this case:

$$\lambda_m(t) \sim \frac{\beta}{t}$$

as $t \to \infty$, which can be shown by direct integration:

$$\begin{aligned}
&\int_0^\infty z f_0(tz)\pi(z)\,dz \\
&= \int_{1/t}^\infty \frac{\beta z}{\Gamma(\alpha+1)t^{\beta+1}z^{\beta+1}} \exp(-z)z^\alpha\,dz \\
&= \frac{\beta}{\Gamma(\alpha+1)t^{\beta+1}} \int_{1/t}^\infty z^{\alpha-\beta}\exp(-z)\,dz \\
&\sim \frac{\Gamma(\alpha-\beta+1)\beta}{\Gamma(\alpha+1)t^{\beta+1}}
\end{aligned}$$

and

$$\int_0^\infty \bar{F}_0(tz)\pi(z)\,dz = \int_0^{1/t} \frac{\exp(-z)z^\alpha}{\Gamma(\alpha+1)}\,dz + \int_{1/t}^\infty \frac{\exp(-z)z^\alpha}{t^\beta z^\beta \Gamma(\alpha+1)}\,dz.$$

As $t \to \infty$, the first integral on the right-hand side is equivalent to

$$\int_0^{1/t} \frac{z^\alpha}{\Gamma(\alpha+1)}\,dz = \frac{1}{t^{\alpha+1}\Gamma(\alpha+2)}$$

and the second integral is equivalent to $\Gamma(\alpha-\beta+1)/\Gamma(\alpha+1)t^\beta$, which decreases more slowly for $\beta \le \alpha$; therefore, the sum of the two integrals is $\Gamma(\alpha-\beta+1)/\Gamma(\alpha+1)t^\beta$. Eventually:

$$\lambda_m(t) \sim \frac{\Gamma(\alpha-\beta+1)\beta}{\Gamma(\alpha+1)t^{\beta+1}} \cdot \frac{\Gamma(\alpha+1)t^\beta}{\Gamma(\alpha-\beta+1)} = \frac{\beta}{t}.$$

If $\beta = \alpha+1$, then

$$\begin{aligned}
&\int_0^\infty z f_0(tz)\pi(z)\,dz \\
&= \frac{\alpha+1}{\Gamma(\alpha+1)t^{\alpha+2}} \int_{1/t}^\infty z^{-1}\exp(-z)\,dz
\end{aligned}$$

and since

$$\int_0^{1/t} z^\alpha\,dz = o(t^{-\alpha-1}) \int_{1/t}^\infty z^{-1}\exp(-z)\,dz,$$

we obtain

$$\begin{aligned}
&\int_0^\infty \bar{F}_0(tz)\pi(z)\,dz \\
&= \frac{1}{\Gamma(\alpha+1)t^{\alpha+1}} \int_{1/t}^\infty z^{-1}\exp(-z)\,dz.
\end{aligned}$$

Therefore

$$\lambda_m(t) \sim \frac{\alpha+1}{t} = \frac{\beta}{t}.$$

and both cases can be combined into one relation

$$\lambda_m(t) \sim \frac{\min(\beta, \alpha+1)}{t}.$$

It can be shown that the same asymptotic relation holds not only for the gamma distribution, but also for any other mixing distribution $\pi(z)$ of the form $\pi(z) = z^\alpha \pi_1(z)$. If $\beta > \alpha+1$, the function $\pi_1(z)$ should be bounded and $\pi_1(0) \neq 0$.

Due to its simplicity, the asymptotic behavior of $\lambda_m(t)$ in the additive hazards model (20.51) does not warrant special attention. As $A(s) = s$ and $\phi(t) = t$, conditions (20.53) and (20.54) of Theorem 20.8 hold and the asymptotic result in (20.55) simplifies to:

$$\lambda_m(t) - \psi'(t) \sim \frac{\alpha+1}{t}.$$

References

20.1 R. Barlow, F. Proschan: *Statistical Theory of Reliability and Life Testing. Probability Models* (Holt, Rinehart and Winston, New York 1975)

20.2 J. D. Lynch: On conditions for mixtures of increasing failure rate distributions to have an increasing failure rate, Probab. Eng. Inform. Sci. **13**, 33–36 (1999)

20.3 J. W. Vaupel, K. G. Manton, E. Stallard: The impact of heterogeneity in individual frailty on the dynamics of mortality, Demography **16**, 439–454 (1979)

20.4 O. O. Aalen: Heterogeneity in survival analysis, Statistics in Medicine **7**, 1121–1137 (1988)

20.5 M. S. Finkelstein: Minimal repair in heterogeneous populations, J. Appl. Probab. **41**, 281–286 (2004)

20.6 Y. Kebir: On hazard rate processes, Naval Res. Log. **38**, 865–877 (1991)

20.7 A. I. Yashin, K. G. Manton: Effects of unobserved and partially observed covariate processes on system failure: A review of models and estimation strategies, Statist. Sci. **12**, 20–34 (1997)

20.8 B. Gompertz: On the nature of the function expressive of the law of human mortality and on a new mode of determining the value of life contingencies, Philos. Trans. R. Soc. **115**, 513–585 (1825)

20.9 M. S. Finkelstein, V. Esaulova: Modeling a failure rate for the mixture of distribution functions, Probab. Eng. Inform. Sci. **15**, 383–400 (2001)

20.10 A. R. Thatcher: The long-term pattern of adult mortality and the highest attained age, J. R. Statist. Soc. A **162**, 5–43 (1999)

20.11 J. R. Carey, P. Liedo, J. W. Vaupel: Slowing of mortality rates at older ages of medfly cohorts, Science **258**, 457–461 (1992)

20.12 M. S. Finkelstein: On some reliability approaches to human aging, Int. J. Reliab. Qual. Safety Eng. **12**, 1–10 (2005)

20.13 P. L. Gupta, R. C. Gupta: Aging characteristics of the Weibull mixtures, Probab. Eng. Inform. Sci. **10**, 591–600 (1996)

20.14 S. Ross: *Stochastic Processes* (Wiley, New York 1996)

20.15 M. Shaked, J. G. Shanthikumar: *Stochastic Orders and Their Applications* (Academic, Boston 1993)

20.16 H. W. Block, J. Mi, T. H. Savits: Burn-in and mixed populations, J. Appl. Probab. **30**, 692–702 (1993)

20.17 H. W. Block, Y. Li, T. H. Savits: Initial and final behavior of failure rate functions for mixtures and functions, J. Appl. Probab. **40**, 721–740 (2003)

20.18 C. A. Clarotti, F. Spizzichino: Bayes burn-in and decision procedures, Probab. Eng. Inform. Sci. **4**, 437–445 (1990)

20.19 N. J. Lynn, N. D. Singpurwalla: Comment: "Burn-in" makes us feel good, Statist. Sci. **12**, 13–19 (1997)

20.20 J. Gurland, J. Sethuraman: How pooling failure data may reverse increasing failure rate, J. Am. Statist. Assoc. **90**, 1416–1423 (1995)

20.21 H. Block, H. Joe: Tail behavior of the failure rate functions of mixtures, Lifetime data analysis **3**, 268–288 (1997)

20.22 M. S. Finkelstein: Why the mixture failure rate bends down with time, South African Statist. J. **39**, 23–33 (2005)

20.23 M. Shaked, F. Spizzichino: Mixtures and monotonicity of failure rate functions. In: *Advances in Reliability*, ed. by N. Balakrishnan, C. R. Rao (Elsevier, Amsterdam 2001) pp. 185–197

20.24 R. Kaas, A. van Heerwaarden, M. Goovaerts: *Ordering of Actuarial Risks* (CAIRE, Brussels 1994)

20.25 J. W. Vaupel, A. I. Yashin: Heterogeneity ruses: some surprising effects of selection on population dynamics, Am. Statist. **39**, 176–185 (1985)

20.26 J. E. Anderson, T. A. Louis: Survival analysis using a scale change random effects model, J. Am. Statist. Assoc. **90**, 669–679 (1995)

20.27 N. H. Bingham, C. M. Goldie, J. L. Teugels: *Regular Variation* (Cambridge Univ. Press, Cambridge 1987)

20.28 M. S. Finkelstein, V. Esaulova: On inverse problem in mixture hazard rates modeling, Appl. Stoch. Models Busin. Ind. **17**, 221–229 (2001)

21. Proportional Hazards Regression Models

The proportional hazards model plays an important role in analyzing data with survival outcomes. This chapter provides a summary of different aspects of this very popular model.

The first part gives the definition of the model and shows how to estimate the regression parameters for survival data with or without ties. Hypothesis testing can be built based on these estimates. Formulas to estimate the cumulative hazard function and the survival function are also provided. Modified models for stratified data and data with time-dependent covariates are also discussed.

The second part of the chapter talks about goodness-of-fit and model checking techniques. These include testing for proportionality assumptions, testing for function forms for a particular covariate and testing for overall fitting.

The third part of the chapter extends the model to accommodate more complicated data structures. Several extended models such as models with random effects, nonproportional models, and models for data with multivariate survival outcomes are introduced.

In the last part a real example is given. This serves as an illustration of the implementation of the methods and procedures discussed in this chapter.

The proportional hazards model has played a pivotal role in survival analysis since it was proposed by *Cox* [21.1]. This model has been widely used in many areas, such as biomedical research and engineering, for assessing covariate effects on the time to some events in the presence of right censoring. For example, when testing the reliability of an electrical instrument, the model can be used to investigate the effects of variables such as humidity, temperature, and voltage on the time to breakdown. Since time constraint might not allow us to observe the failure of every experimental unit, for some units we only know that failure did not occur up to the end of study, which is the censoring event.

Let T be the failure time, C be the censoring time, and $\boldsymbol{Z} = \{\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_p\}^T$ be a p-dimensional vector of covariates. Throughout this chapter the covariate vector $\boldsymbol{Z}$ is assumed to be time-independent, although it is straightforward to extend the theory to time-varying covariates. The failure time T might not always be observed due to censoring, and what we actually observe

are $X = \min(T, C)$, the smaller of the failure time and the censoring time, and $\Delta = I\,(T \le C)$, the indicator that failure has been observed. The dataset obtained from a failure-time study consists of n independent realizations of the triplet $(X, \boldsymbol{Z}, \Delta)$. It is usually assumed that the censoring is noninformative in that, given $\boldsymbol{Z}$, the failure and the censoring times are independent. Let $P\,(T > t \mid \boldsymbol{Z})$ be the conditional survival function, and the conditional hazard function is defined as

$$\lambda\,(t \mid \boldsymbol{Z}) = \lim_{\Delta t \downarrow 0} \frac{1}{\Delta t} P\,(t \le T < t + \Delta t \mid T \ge t, \boldsymbol{Z})\ ,$$

which is the instantaneous rate of failure at time t, given that failure has not occurred before t and the covariate vector $\boldsymbol{Z}$.

There are many ways to model the relationship between the failure time and the covariates. The proportional hazards model specifies

$$\lambda\,(t \mid \boldsymbol{Z}) = \lambda_0(t) \exp\left(\beta^T \boldsymbol{Z}\right)\ , \tag{21.1}$$

where $\lambda_0(t)$ is an unknown baseline hazard function corresponding to $\boldsymbol{Z} = (0, \cdots, 0)$, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the vector of regression coefficients.

This method does not assume a parametric distribution for the failure times, but rather assumes that the effects of the different variables on the time to failure are constant over time and are multiplicative on the hazard. The model is called the proportional hazards model since the ratio of hazards of any two experimental units is always a constant:

$$\frac{\lambda\,(t \mid z)}{\lambda\,(t \mid z')} = \frac{\lambda_0(t) \exp\left(\beta^T z\right)}{\lambda_0(t) \exp\left(\beta^T z'\right)} = \exp\left[\beta^T (z - z')\right]\ ,$$

where z and z' are the respective covariate values of the two units. This quantity is often referred to as the hazard ratio or relative risk.

The interpretation of the parameter β is similar to that in other regression models. For example, $\exp(\beta_1)$ is the hazard ratio of two study units whose values of the first covariate differ by 1 and whose values of any other covariate are the same. Usually, the goal is to make inferences about β or a subset of β to see whether a certain covariate has an effect on the survival rate or not. The baseline hazard $\lambda_0(\cdot)$ is treated as a nuisance parameter function. The proportional hazards model is considered a semiparametric model, in the sense that $\lambda_0(\cdot)$ is an infinite-dimensional parameter.

The semiparametric proportional hazards model includes the parametric Weibull model as a special case. To see this, for the Weibull distribution with density $f(t) = \alpha\lambda t^{\alpha-1} \exp(-\lambda t^{\alpha})$ and survival function $S(t) = \exp(-\lambda t^{\alpha})$, parameterize the parameter λ as $\lambda = \lambda^{'} \exp(\beta^T \boldsymbol{Z})$, then the hazard of failure given $\boldsymbol{Z}$ is

$$\lambda(t|\boldsymbol{Z}) = \lambda_0(t) \exp\left(\beta^T \boldsymbol{Z}\right)\ ,$$

where $\lambda_0(t) = \alpha\lambda^{'} t^{\alpha-1}$ is a function with two parameters, instead of the unspecified $\lambda_0(\cdot)$ in the case of the proportional hazards model. It can also be shown that the Weibull model is also a special case of the semiparametric accelerated failure-time model. In fact, the Weibull model is the most general parametric model that has both the proportional hazards and the accelerated failure-time properties. See Chapt. 12 of *Klein* and *Moeschberger* [21.2] for a detailed discussion.

21.1 Estimating the Regression Coefficients β

The partial likelihood method was introduced by *Cox* [21.3] to estimate the regression parameters β in the proportional hazards model for failure times with possible right censoring. We will first focus on the case when all failure times are distinct. When the failure time follows a continuous distribution, it is very unlikely that two subjects would fail at the same time. In reality, however, the measured time always has a discrete distribution, since it can only take values in a finite set of numbers. Thus tied failure times could happen in a real study, and special attention is needed in this situation.

21.1.1 Partial Likelihood for Data with Distinct Failure Times

Now suppose there is no tie among the failure times. Let $t_1 < \cdots < t_N$ denote the N ordered times of observed failures and let (j) denote the label of the individual that fails at t_j. Let $\mathcal{R}_j$ be the risk set at time t_j, i. e. $\mathcal{R}_j = \{i : X_i \ge t_j\}$.

The partial likelihood for the model (21.1) is defined as

$$\prod_{j=1}^{N} \frac{\exp\left(\beta^T Z_{(j)}\right)}{\sum_{i \in \mathcal{R}_j} \exp\left(\beta^T Z_i\right)}\ , \tag{21.2}$$

and the log partial likelihood is then

$$L(\beta)=\sum_{j=1}^{N}\left\{\beta^T Z_{(j)}-\log\left[\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)\right]\right\}.$$

The maximum partial likelihood estimate of β, $\hat{\beta}$, as proposed by *Cox* [21.3], is found by solving the score equation

$$U(\beta)=0\,,$$

where $U(\beta)=\partial L(\beta)/\partial\beta$.

The information matrix, defined as the negative of the second derivative matrix of the log likelihood, is given by

$$I(\beta)=-\frac{\partial U(\beta)}{\partial\beta}=\sum_{j=1}^{N}\left[\frac{\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)Z_i^{\otimes 2}}{\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)}-\left(\frac{\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)Z_i}{\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)}\right)^{\otimes 2}\right],$$

where $\boldsymbol{a}^{\otimes 2}=\boldsymbol{a}\boldsymbol{a}^T$ for any vector $\boldsymbol{a}$.

It can be shown that $\hat{\beta}$ is a consistent estimator for β, and $nI^{-1}(\hat{\beta})$ is a consistent estimator for the covariance matrix of $n^{1/2}(\hat{\beta}-\beta)$, where n is the number of all subjects, censored or uncensored. Thus for large samples, $\hat{\beta}$ has an approximately normal distribution with mean β and variance–covariance matrix $I^{-1}(\hat{\beta})$.

21.1.2 Partial Likelihood for Data with Tied Failure Times

In the previous section, we defined the partial likelihood for data with distinct failure times. Now we want to give several alternative partial likelihoods for data with ties between failure times.

Suppose there are N distinct observed failure times $t_1<\cdots<t_N$, and at each time $t_j(1\le j\le N)$ there are d_j observed failures. Let $\mathcal{D}_j$ be the set of all individuals who die at time t_j. Let $\mathcal{R}_j$ be the risk set at time t_j, i. e. $\mathcal{R}_j=\{i: X_i\ge t_j\}$.

When there are many ties in the data, the computation of maximum partial likelihood estimates, though still feasible, becomes time-consuming. For this reason, approximations to the partial likelihood function are often used. Two commonly employed approximations are due to Breslow and to Efron.

Breslow [21.4] suggested the following log partial likelihood for data with ties among failure times

$$L_{\mathrm{B}}(\beta)=\sum_{j=1}^{N}\left\{\beta^T\sum_{l\in D_j}Z_l-d_j\log\left[\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)\right]\right\}.$$

This approximation works well when there are not many ties. Another approximation of the log partial likelihood is given by *Efron* [21.5]

$$L_{\mathrm{E}}(\beta)=\sum_{j=1}^{N}\left\{\beta^T\sum_{l\in D_j}Z_l-\sum_{k=1}^{d_j}\log\left[\sum_{i\in\mathcal{R}_j}\exp\left(\beta^T Z_i\right)-(k-1)/d_j\sum_{i\in\mathcal{D}_j}\exp\left(\beta^T Z_i\right)\right]\right\}.$$

Breslow's method is easy to use and is therefore more popular, but Efron's approximation is generally the more accurate of the two. Also both likelihoods reduce to the partial likelihood when there is no tie.

21.2 Estimating the Hazard and Survival Functions

The cumulative baseline hazard function $\Lambda_0(t)=\int_0^t\lambda_0(u)\,\mathrm{d}u$ can be estimated by *Breslow* [21.6]

$$\hat{\Lambda}_0(t)=\sum_{j:\,t_j\le t}\frac{\delta_j}{\sum_{i\in R_j}\exp(\hat{\beta}^T Z_i)}\,,$$

where $\delta_j=I(T_j\le C_J)$. Note that $\hat{\Lambda}_0$ is a right-continuous step function with jumps at the observed failure times, and it is often referred to as the Breslow estimator. In the case of tied events, each of the subjects in a tie contributes its own term to the sum, and this term is the same for all subjects who failed at the specific time. This estimator can also be derived through a profile likelihood approach (*Johansen* [21.7], *Klein* and *Moeschberger* [21.2]). The baseline survival function $S_0(t)=\exp[-\Lambda_0(t)]$ can thus be estimated by $\hat{S}_0(t)=\exp[-\hat{\Lambda}_0(t)]$. The estimated survival function of an individual with covariate value z is given by

$$\hat{S}(t\mid z)=\exp\left[-\hat{\Lambda}_0(t)\mathrm{e}^{\hat{\beta}^T z}\right].$$

21.3 Hypothesis Testing

Without loss of generality, assume that we are interested in hypothesis testing involving only the first q components of the regression parameter β. Write $\beta = \left(\beta_1^T, \beta_2^T\right)^T$, where β_1 is of dimension q and β_2 is of dimension $(p-q)$. For testing the null hypothesis $\beta_1 = \beta_{01}$ against the alternative $\beta_1 \neq \beta_{01}$ for any fixed β_{01} in the presence of the unknown parameters β_2, there are three types of tests: the likelihood ratio test, the Wald test, and the score test. This type of test with $\beta_{01} = 0$ is often used in model selection procedures, testing whether a given model can be improved by including a certain additional covariate or covariate combinations.

21.3.1 Likelihood Ratio Test

The test statistic for the likelihood ratio test is given by

$$TS_{\text{LR}} = 2\left[\log L(\hat{\beta}) - \log L(\tilde{\beta})\right],$$

where $\tilde{\beta} = (\beta_{01}, \tilde{\beta}_2^T)^T$ and $\tilde{\beta}_2$ maximizes $L(\beta)$ when β_1 is fixed at β_{01}. Under the null hypothesis, the asymptotic distribution of TS_{LR} is χ_q^2.

21.3.2 Wald Test

Let $\hat{\beta} = \left(\hat{\beta}_1^T, \hat{\beta}_2^T\right)$ denote the usual maximum partial likelihood estimate of the full parameter vector $\boldsymbol{\beta} = \left(\beta_1^T, \beta_2^T\right)$, and partition the inverse of the information matrix as

$$\mathbf{I}^{-1}(\beta) = \begin{pmatrix} \mathbf{I}^{11}(\beta) & \mathbf{I}^{12}(\beta) \\ \mathbf{I}^{21}(\beta) & \mathbf{I}^{22}(\beta) \end{pmatrix},$$

where $\mathbf{I}^{11}(\beta)$ is a $q \times q$ matrix. The test statistic for the Wald test is given by

$$TS_{\text{wald}} = (\hat{\beta}_1 - \beta_{01})^T \times \mathbf{I}^{11}(\hat{\beta})^{-1}(\hat{\beta}_1 - \beta_{01}).$$

Under the null hypothesis, the asymptotic distribution of TS_{wald} is χ_q^2.

21.3.3 Score Test

Let $S_1(\beta)$ denote the vector of the first q components of the score function $S(\beta)$. The test statistic for the score test is

$$TS_{\text{score}} = S_1(\tilde{\beta})^T \times \mathbf{I}^{11}(\tilde{\beta}) S_1(\tilde{\beta}),$$

where $\tilde{\beta}$ and $\mathbf{I}_{11}(\beta)$ are defined as before. Again, the large sample distribution of the test statistic under the null hypothesis is χ_q^2.

21.4 Stratified Cox Model

The proportional hazards model can be stratified to account for heterogeneity in the baseline hazards. To achieve this, the subjects are divided into several groups with distinct baseline hazard functions and a common vector of regression coefficients β, and proportional hazards are assumed within each stratum. For a subject with covariate $\boldsymbol{Z}$ in the k-th stratum, let the hazard at time t be

$$\lambda(t \mid Z) = \lambda_k(t) \exp\left(\beta^T Z\right).$$

Within each stratum a partial likelihood function can be defined as in (21.2), and the partial likelihood for the stratified Cox model is defined as the sum of the partial likelihood functions for all strata.

21.5 Time-Dependent Covariates

The Cox model can be extended to include time-dependent covariates.

Let $\boldsymbol{Z}(t)$ be a covariate vector measured at time t. Again we assume that the censoring is noninformative in that the failure time T and the censoring time C are conditionally independent, given the history of the covariate vector $\boldsymbol{Z}^*(X)$, where $\boldsymbol{Z}^*(t) = \{Z(u) : 0 \leq u < t\}$ for any $0 \leq t \leq X$. The dataset $\left\{\left[X_i, Z_i^*(X_i), \Delta_i\right] : i = 1, \cdots, n\right\}$ is an i.i.d. sample of $\left\{\left[X, Z^*(X), \Delta\right]\right\}$. Using similar notations as in Sect. 21.1, the hazard function for T is defined as

$$\begin{aligned}\lambda\left[t \mid \boldsymbol{Z}^*(t)\right] &= \lim_{\Delta t \downarrow 0} \frac{1}{\Delta t} P\left[t \leq T < t + \Delta t \mid T \geq t, \boldsymbol{Z}^*(t)\right] \\ &= \lambda_0(t) \exp\left[\beta^T z(t)\right].\end{aligned}$$

The partial likelihood is then defined as

$$\prod_{j=1}^{N} \frac{\exp\left[\beta^T Z_{(j)}(X_{(j)})\right]}{\sum_{i\in\mathcal{R}_j} \exp\left[\beta^T Z_i(X_{(j)})\right]},$$

where, like in Sect. 21.2, (j) denotes the label of the subject with the j-th earliest observed failure time, and R_j denotes the corresponding risk set.

Though the extension of the model and partial likelihood is simple, the validity of the underlying assumption and the computation of the partial likelihood are quite complicated in practice. Extra care is needed in dealing with time-dependent covariates.

Kalbfleisch and *Prentice* [21.8] pointed out that there are two types of time-dependent covariates: external covariates, whose value do not depend on the failure process, and internal covariates, which usually carry information about the failure process. For example, when studying how long an experimental unit remains functioning, the air humidity rate is essentially external to the units work duration. But a patient's daily blood pressure is an internal time-dependent covariate, since it carries information about the health status and hence the failure time of the patient. Though the definition of the hazard function and the construction of the partial likelihood apply to both type of time-dependent covariates, it is not possible to estimate the conditional survival function when there are internal covariates.

Another problem concerns the measurement of time-dependent covariates. The calculation of partial likelihood requires that the values of any unit's time-dependent covariates be available at all failure times when it is still at risk. This cannot be achieved in general, since we can never know the failure times in advance and the information on the covariates are usually collected at predetermined time points.

One way to deal with incomplete history of time-dependent covariates is imputation, and there are several possible ways to impute the intermittent values. An ad hoc approach commonly used in practice, referred to as the LVCF method, is to impute the missing covariate at a certain time point with the nearest previous observation of the same unit. Other nonparametric approaches, like the smoothing methods, can be used to estimate the unobserved part of the time-dependent data.

If one can make the assumption that the time-dependent covariates follow certain models, for example, linear mixed effects models, several strategies can be applied, including regression calibration methods [21.9–11], joint likelihood methods, [21.12–14], and conditional score methods, [21.15].

One also needs to be aware of the possible informative censoring due to the fact that the time-dependent covariates are truncated by failure [21.16]. *Allison* [21.17] suggested avoiding using observations after the failure time, unless one is dealing with external time-dependent covariates. Failure to take into account these problems can lead to biased parameter estimates.

21.6 Goodness-of-Fit and Model Checking

21.6.1 Tests of Proportionality

A key assumption of the Cox proportional hazard model is the proportionality of the hazards. Note that the hazard function for an individual depends on the covariate values and the value of the baseline hazard. For any two individuals, it is easy to see from (21.1) that the ratio of the hazards over time will be constant. The validity of this assumption needs to be checked. Various methods can be used for this purpose.

One graphical approach to check the proportionality assumption is the log–log survival plot. We first divide all study units into several groups according to the covariate values, and then estimate the survival function within each group using the *Kaplan–Meier* method, [21.18]. If the proportional hazard assumption is satisfied, plotting the transformed survival functions, $\log\left[-\log S(t)\right]$, would result in parallel curves. This method works well with categorical covariates without many levels. For continuous predictors, one may first divide them into a few categories based on quantiles or other grouping criterion. For categorical covariates with many levels, one may want to combine them into fewer groups.

Other graphic approaches includes *Andersen* plots [21.19], *Arjas* plots [21.20], and the use of plots based on the score residuals or Schoenfeld residuals. The details of these approaches can be found in *Klein* and *Moeschberger* [21.2], Chapt. 11.

Alternatively, one can check the proportionality assumption by adding a time-dependent covariate to the model. The time-dependent covariate can be defined

as the interaction of a time-independent covariate and a function of time. If the time-dependent covariate is significant in the Cox model, it could be concluded that the effect of the time-independent covariate varies with time, and the proportional hazards assumption is violated.

21.6.2 Test of the Functional Form of a Continuous Covariate

Next we want to examine the functional form of a given covariate on the survival and see whether any transformation of the original covariate is needed. One method is based on martingale residuals.

Let $\hat{\beta}$ be the estimated regression coefficient and $\hat{\Lambda}_0(\cdot)$ be the Breslow estimator of the cumulative baseline hazard function, the martingale residual for the i-th subject is defined as

$$\hat{M}_i = \delta_i - \hat{\Lambda}_0(X_i)\exp\left(\hat{\beta}^T Z_i\right) = \delta_i - \hat{\Lambda}_i(X_i)\,.$$

It can be shown that when sample size is large, expectation of $\hat{M}_i$ is close to 0 and, for $i \neq j$, $\mathrm{cov}(\hat{M}_i, \hat{M}_j)$ is also close to 0. We also have $\sum \hat{M}_i = 0$.

When the functional form of a specific covariate needs to be examined, we first fit a Cox model with all covariates excluding the covariate to be investigated, and then plot the martingale residuals against the excluded covariate. If the plot shows an approximately linear trend, no transformation is needed and the untransformed covariate can be included in the model with the other covariates. If, however, there appears to be a certain pattern, a proper transformation of the original covariate might be needed.

Nonparametric methods can also be used to explored the nature of covariate effects on survival. Local likelihood or penalized likelihood methods with different smoothing methods have been proposed to estimate the functional form of a single covariate or a linear combination of covariates in the survival model [21.21–25].

21.6.3 Test for the Influence of Individual Observation

The influence of individual observation may be studied by the use of score residuals. We can fit the Cox model with and without the i-th observation of the data sample, and obtain the estimates $\hat{\beta}$ and $\hat{\beta}_{(i)}$, respectively. If $\hat{\beta} - \hat{\beta}_{(i)}$ is close to zero the individual observation has little influence on the estimate. We can plot this difference against the observation number to identify those influential observations.

21.6.4 Test for the Overall Fit

The overall fit of the model can be assessed using the Cox–Snell residuals, which are defined as

$$r_i = \hat{\Lambda}_0(X_i)\exp\left(\hat{\beta}^T Z_i\right), \qquad i = 1, \cdots, n\,,$$

where $\hat{\Lambda}$ is the Breslow estimator for the cumulative baseline hazard function and $\hat{\beta}$ are the estimated regression coefficients. It can be shown that, when there is no censoring and the true values of the parameters are known, $\Lambda_0(T_i)\exp\left(\beta^T Z_i\right)$ follows an exponential distribution with unit rate. Thus we can treat $\{r_i, i = 1, \ldots, n\}$ as a possibly right-censored sample of failure times from the unit exponential distribution, which has a constant hazard rate $\lambda_{\mathrm{r}}(r) = 1$ and a cumulative hazard function $\Lambda_{\mathrm{r}}(r) = r$. The failure indicator δ_i for X_i can also serve as the failure indicator for r_i. The Nelson–Aalen estimator can be used to estimate Λ_{r}:

$$\hat{\Lambda}_{\mathrm{r}}(r) = \sum_{i:\, r_i \le r} \frac{\delta_i}{\#\{r_j : r_j \ge r_i\}}\,,$$

where # counts the number of elements in a set. If the model is correct, the function $\hat{\Lambda}_{\mathrm{r}}$ would be close to the straight line $\Lambda_{\mathrm{r}}(r) = r$. Thus, plotting $\hat{\Lambda}_{\mathrm{r}}(r_i)$ against r_i will provide an assessment of the departure from the model assumptions.

21.6.5 Test of Time-Varying Coefficients

Schoenfeld residuals [21.26] can be used to test for time-varying coefficients in the Cox model. Let $Y_i(t) = I(X_i \ge t)$, be the indicator of whether unit i is still at risk at time t. For a subject who fails at time t_i, the vector of Schoenfeld residuals is defined as

$$Z_i(t_i) - \bar{Z}(t_i)\,,$$

where $\bar{Z}(t)$ is defined as

$$\bar{Z}(t) = \frac{\sum_{j=1}^{n} Y_j(t) Z_j \exp\left(\hat{\beta}^T Z_j\right)}{\sum_{j=1}^{n} Y_j(t)\exp\left(\hat{\beta}^T Z_j\right)}\,,$$

which is a weighted average of the covariate $\boldsymbol{Z}$ over all individuals at risk at time t.

Under the proportional hazards assumption, the Schoenfeld residuals should be independent of time. Therefore, a clear pattern of the Schoenfeld residuals over time implies a departure from the model assumption.

For the j-th covariate, a time-varying coefficient can be expressed as

$$\beta_j(t) = b_{0j} + b_{1j} f_j(t) ,$$

where $f_j(t)$ is a known function of time. *Grambsch* and *Therneau* [21.27] showed that the scaled Schoenfeld residuals from a Cox model have a mean of approximately $b_{1j} f_j(t)$ for the j-th covariate at time t. So the plot of the scaled Schoenfeld residuals against the event times can be used to check whether the coefficient of f_j is zero.

21.6.6 Test for a Common Coefficient Across Different Groups

To test whether the effect of a covariate on failure is identical across different groups of study units, a stratified analysis can be utilized. First, a stratified Cox model is fitted with different regression coefficients for different groups. Then another stratified Cox model is fitted with the same coefficient for all groups. The difference between the log likelihoods, which has an approximate χ^2 distribution, can be used to test for any heterogeneity in the covariate effect.

21.7 Extension of the Cox Model

21.7.1 Cox Model with Random Effects

Similarly to the case of a linear model, random effects can be added to the proportional hazards model to handle clustered or heterogeneous survival data [21.28]. Let

$$\lambda(t \mid \boldsymbol{Z}) = \lambda_0(t) \exp\left(\beta^T \boldsymbol{Z} + b^T \boldsymbol{W}\right) ,$$

where b is the random effect of the covariate vector $\boldsymbol{W}$ on survival. This model allows for a multivariate random effect with known distribution. Maximum likelihood estimates of the regression parameters, the variance components and the baseline hazard function can be obtained via the expectation maximization (EM) algorithm.

21.7.2 Nonproportional Models

Several models can be applied when one suspects that the proportionality assumption does not hold for a certain dataset. The most common among them are frailty models and cure-rate models.

Frailty models can be used to account for individual randomness in an experiment. An unobserved random variable W is added to a Cox proportional hazards model, which is assumed to follow a known distribution. The effect of W is multiplicative on the hazard: given W, the hazard rate is given by

$$\lambda(t \mid \boldsymbol{W}, \boldsymbol{Z}) = \lambda_0(t) W \exp\left(\beta^T \boldsymbol{Z}\right) .$$

With a common choice for the distribution of W, Gamma$(1, \theta)$, the marginal hazard rate given the covariate vector $\boldsymbol{Z}$ is

$$\lambda(t \mid \boldsymbol{Z}) = \frac{\lambda_0(t) \exp\left(\beta^T \boldsymbol{Z}\right)}{1 + \theta \exp\left(\beta^T \boldsymbol{Z}\right) \Lambda_0(t)} .$$

Another commonly used model for nonproportional hazards is the cure-rate model. This assumes that there are two sub-populations: cured subjects and uncured subjects. Suppose the proportion of the cured subjects is π and that of the uncured subjects is $1 - \pi$. The survival probability at t given covariates $\boldsymbol{Z}$ is then given by

$$S(t \mid \boldsymbol{Z}) = \pi S_1(\boldsymbol{Z}) + (1 - \pi) S_2(t \mid \boldsymbol{Z}) ,$$

where $S_1(\boldsymbol{Z})$ is the probability of being cured and $S_2(t \mid \boldsymbol{Z})$ represents the survival function of the uncured,

Table 21.1 Data table for the example

Time to breakdown (s)				
45 kV	40 kV	35 kV	30 kV	25 kV
1	1	30	50	521
1	1	33	134	2517
1	2	41	187	4056
2	3	87	882	12553
2	12	93	1448	40290
3	25	98	1468	50560+
9	46	116	2290	52900+
13	56	258	2932	67270+
47	68	461	4138	83990
50	109	1182	15750	85500+
55	323	1350	29180+	85700+
71	417	1495	86100+	86420+

+ represents censoring

which is modeled as in a proportional hazards model. The cure-rate model can be used to fit failure-time data when the right tail of the survival function looks like a plateau.

21.7.3 Multivariate Failure Time Data

Sometimes an experimental unit might experience multiple failures, which could be of the same nature (e.g., same event recurring over time), or of different nature (e.g., distinct types of problems). Also, in some situations there are clustering of study units such that failure times within the same cluster are expected to be correlated. For example, pieces of equipment in the same factory might behave similarly to those in another factory.

Various methods have been developed for these kinds of multivariate failure-time data. See, for example, [21.29–33].

21.8 Example

We now use a dataset from the book by *Wayne* [21.34] as an illustration. The data came from an experiment on testing the fatigue limit for two steel specimens in two forms under different stress ratios. The data are shown in Table 21.1.

The fitting result is shown in Table 21.2. We can see that voltage has a negative effect on the failure rate. The higher the voltage is, the shorter the time to break down will be. One unit increase in voltage results in an hazard ratio of 1.27 (95% confidence interval = 1.19–1.36).

The survival function estimates for five groups of voltage are displayed in Fig. 21.1. The graph shown in Fig. 21.2 of the graph of log[-log(survival)] versus the log of survival time results in parallel straight lines, so the proportional hazard assumption is satisfied. The

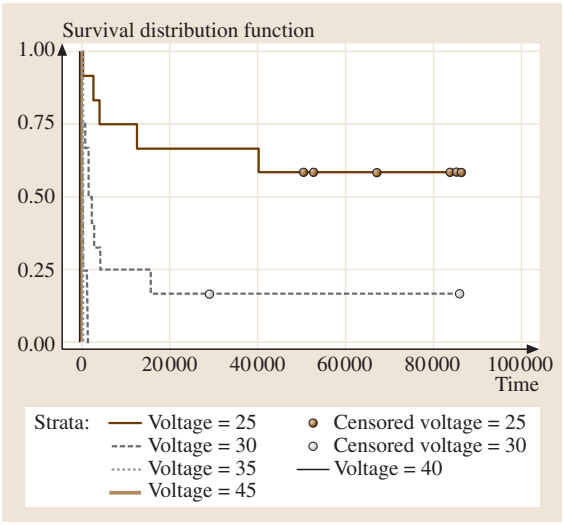


Fig. 21.1 Estimated survival function

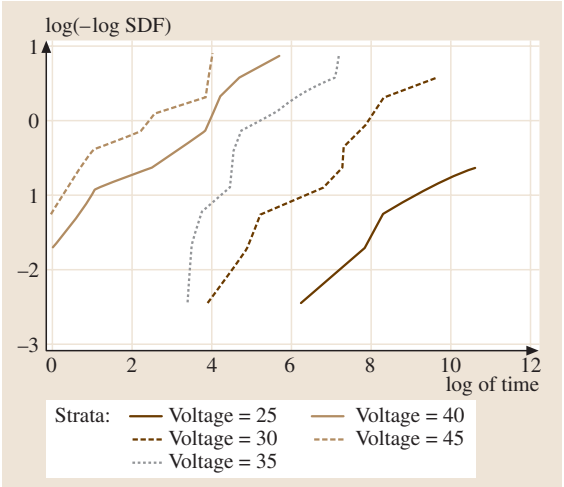


Fig. 21.2 Checking the proportional hazard assumption

Table 21.2 Model fitting result

	coef	exp(coef)	se(coef)	z	p
Voltage	0.241	1.27	0.0339	7.12	1.1e-012
	exp(coef)	exp(-coef)	lower 0.95	upper 0.95	
Voltage	1.27	0.786	1.19	1.36	
Rsquare = 0.68 (max possible = 0.997)					
Likelihood ratio test = 68.3				on 1 df,	p = 1.11e-016
Wald test = 50.7				on 1 df,	p = 1.07e-012
Score (logrank) test = 68.1				on 1 df,	p = 1.11e-016

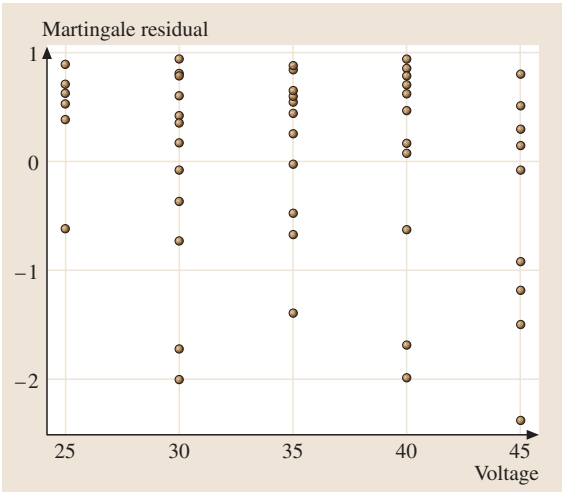


Fig. 21.3 Martingale residual plot

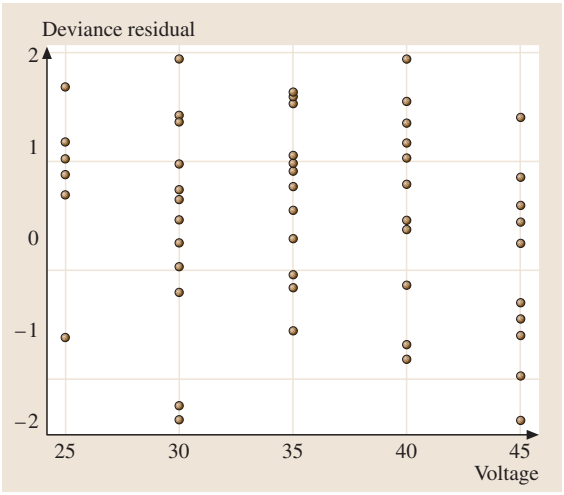


Fig. 21.4 Deviance residual plot

martingale residual plot is shown in Fig. 21.3 and the deviance residual plot is shown in Fig. 21.4. There is no indication of a lack of fit of the model to individual observations.

References

21.1 D. R. Cox: Regression models, life-tables (with discussion), J. R. Stat. Soc. B **34**, 187–220 (1972)

21.2 J. P. Klein, M. L. Moeschberger: *Survival Analysis: Techniques for Censored and Truncated Data* (Springer, Berlin Heidelberg New York 1997)

21.3 D. R. Cox: Partial likelihood, Biometrika **62**, 269–276 (1975)

21.4 N. E. Breslow: Covariance analysis of censored survival data, Biometrics **30**, 89–99 (1974)

21.5 B. Efron: The efficiency of Cox's likelihood function for censored data, J. Am. Stat. Assoc. **72**, 557–565 (1977)

21.6 N. E. Breslow: Contribution to the discussion of the paper by D. R. Cox, J. R. Stat. Soc. B **34**, 187–220 (1972)

21.7 S. Johansen: An extension of Cox's regression model, Int. Stat. Rev. **51**, 258–262 (1983)

21.8 J. D. Kalbfleisch, R. L. Prentice: *The Statistical Analysis of Failure Time Data* (Wiley, New York 1980)

21.9 Y. Pawitan, S. Self: Modeling disease marker processes in AIDS, J. Am. Stat. Assoc. **83**, 719–726 (1993)

21.10 U. G. Dafni, A. A. Tsiatis: Evaluating surrogate markers of clinical outcome when measured with error, Biometrics **54**, 1445–1462 (1998)

21.11 A. A. Tsiatis, V. Degruttola, M. S. Wulfsohn: Modeling the relationship of survival to longitudinal data measured with error: applications to survival, CD4 counts in patients with AIDS, J. Am. Stat. Assoc. **90**, 27–37 (1995)

21.12 C. J. Faucett, D. C. Thomas: Simultaneously modeling censored survival data, repeatedly measured covariates: a Gibbs sampling approach, Stat. Med. **15**, 1663–1685 (1996)

21.13 M. S. Wulfsohn, A. A. Tsiatis: A joint model for survival, longitudinal data measured with error, Biometrics **53**, 330–339 (1997)

21.14 R. Henderson, P. Diggle, A. Dobson: Joint modelling of longitudinal measurements, event time data, Biostat. **4**, 465–480 (2000)

21.15 A. A. Tsiatis, M. Davidian: A semiparametric estimator for the proportional hazards model with longitudinal covariates measured with error, Biometrika **88**, 447–458 (2001)

21.16 M. C. Wu, R. J. Carroll: Estimation, comparison of changes in the presence of informative right censoring by modeling the censoring process, Biometrics **44**, 175–188 (1988)

21.17 P. D. Allison: *Survival Analysis Using the SAS System: A Practical Guide* (SAS Institute, Cary 1995)

21.18 E. L. Kaplan, P. Meier: Nonparametric estimation from incomplete observations, J. Am. Stat. Assoc. **53**, 457–481 (1958)

21.19 P. K. Andersen: Testing goodness of fit of Cox's regression and life model, Biometrics **38**, 67–77 (1982)

21.20 E. A. Arjas: Graphical method for assessing goodness of fit in Cox's proportional hazards model, J. Am. Stat. Assoc. **83**, 204–212 (1988)
21.21 R. Tibshirani, T. Hastie: Local likelihood estimation, J. Am. Stat. Assoc. **82**, 559–567 (1987)
21.22 L. A. Sleeper, D. P. Harrington: Regression splines in the Cox model with application to covariate effects in liver disease, J. Am. Stat. Assoc. **85**, 941–949 (1990)
21.23 R. Gentleman, J. Crowley: Local full likelihood estimation for the proportional hazards model, Biometrics **47**, 1283–1296 (1991)
21.24 J. Fan, I. Gijbels, M. King: Local likelihood, local partial likelihood in hazard regression, Ann. Stat. **25**, 1661–1690 (1997)
21.25 W. Wang: Proportional hazards regression with unknown link function, time-dependent covariates, Stat. Sin. **14**, 885–905 (2004)
21.26 D. Schoenfeld: Partial residuals for the proportional hazards regression model, Biometrika **69**, 239–241 (1982)
21.27 P. M. Grambsch, T. M. Therneau: Proportional hazards tests, diagnostics based on weighted residuals, Biometrika **81**, 515–526 (1994)
21.28 D. J. Sargent: A general framework for random effects survival analysis in the Cox proportional hazards setting, Biometrics **54**, 1486–1497 (1998)
21.29 R. L. Prentice, B. J. Williams, A. V. Peterson: On the regression analysis of multivariate failure time data, Biometrika **68**, 373–379 (1981)
21.30 L. J. Wei, D. Y. Lin, L. Weissfeld: Regression analysis of multivariate incomplete failure time data by modeling marginal distribution, J. Am. Stat. Assoc. **84**, 1065–1073 (1989)
21.31 C. F. Spiekerman, D. Y. Lin: Marginal regression models for multivariate failure time data, J. Am. Stat. Assoc. **93**, 1164–1175 (1998)
21.32 D. Y. Lin, L. J. Wei, I. Yang, Z. Ying: Semiparametric regression for the mean, rate functions of recurrent events, J. R. Stat. Soc. B **62**, 711–730 (2000)
21.33 P. K. Andersen, R. D. Gill: Cox's regression model counting process: a large sample study, Ann. Stat. **10**, 1100–1120 (1982)
21.34 N. Wayne: *Accelerated Testing: Statistical Models, Test Plans, And Data Analysis* (Wiley, New York 1990)

22. Accelerated Life Test Models and Data Analysis

Today's consumers demand high quality and reliability in the products they buy. Accelerated life tests (ALT) are commonly used by manufacturers during product design to obtain reliability information on components and subsystems in a timely manner. The results obtained at high levels of the accelerating variables are then extrapolated to provide information about the product life under normal use conditions.

The introduction and Section 22.1 describe the background and motivations for using accelerated testing. Sections 22.2 and 22.3 discuss statistical models for describing lifetime distributions in ALT. Commonly used ALT models have two parts: (a) a statistical distribution at fixed levels of the accelerated variable(s); and (b) a functional relationship between distribution parameters and the accelerating variable(s). We describe relationships for accelerating variables, such as use rate, temperature, voltage, and voltage rate. We also discuss practical guidelines and potential problems in using ALT models. Section 22.4 describes and illustrates a strategy for analyzing ALT data. Both graphical and numerical methods are discussed for fitting an ALT model to data and for assessing its fit. These methods are thoroughly illustrated by fitting an ALT model with a single accelerating variable to data obtained from an actual ALT experiment. Extrapolation of the results at accelerated levels to normal use levels is also discussed. Section 22.5 presents statistical analysis of a wider variety of ALT data types that are encountered in practice. In particular, the examples involve ALTs with interval censoring and with two or more accelerating variables. Section 22.6 discusses practical considerations for interpreting statistical analysis of ALT data. This Section emphasizes the important role of careful planning of ALT to produce useful results. Section 22.7 discusses other kinds of accelerated tests often conducted in practice. Brief descriptions of each and specific applications in industry are also provided. Section 22.8 reviews some of the potential pitfalls the practitioner of accelerated testing may face. These are described within practical situations, and strategies for avoiding them are presented. Section 22.9 lists some computer software packages that are useful for analyzing ALT data.

Rapid developments in technology, consumer demand for highly reliable products, and competitive markets have placed pressure on manufacturers to deliver products with high quality and reliability. Similarly, there is strong pressure to shorten the concept-to-market cycle time, subject to the constraint of having a product with high reliability. Achieving these goals requires more attention to reliability early in the product design process and has lead to such programs as *design for reliability* and *design for six sigma.* As a result, there has been an increased need for upstream tests of product materials and components, often followed by subsequent testing of subsystems and even complete systems.

Customers may expect products to work properly for years or even decades. For instance, a refrigerator might be expected to work without failure for at least 10 years. Testing under normal operating conditions for reasonable lengths of time is unlikely to result in any failures and thus little information about product reliability. Hence, manufacturers perform accelerated tests (AT) on their products by exposing them to harsher conditions to generate failures and useful reliability information more quickly.

Depending on the nature of the product, life tests are accelerated by increasing product use rate or exposing the product to higher levels of accelerating variables such as temperature, pressure, and voltage. It is customary to fit a statistical model to the AT data and then extrapolate to use conditions to characterize the product's long-term performance. It is desirable that the statistical model for ATs be based on physical/chemical theory that can be used to justify the extrapolation. Operationally, however, detailed knowledge of theory relating the accelerating variables to life is not always available and the model is chosen on the basis of previous experience in similar situations.

Due to space constraints, we provide a limited discussion and description of the models and methods used in AT. For those actually involved in AT and for those who want more information, we highly recommend *Nelson* [22.1]. Other useful books with information that will complement this chapter include *Tobias* and *Trindade* [22.2] and *Meeker* and *Escobar* [22.3].

22.1 Accelerated Tests

22.1.1 Types of Accelerated Tests

During product design, manufacturers perform experiments to obtain timely information on material properties and component durability. Other experiments involving prototype systems and subsystems are used to help make product and process design decisions that will improve product robustness. Experiments are also run to support decision making in production process design. Tests during the production stage include certification of components, burn-in tests, and tests designed to monitor the production process over time. For further discussion of these issues, see *Meeker* and *Hamada* [22.4] and *Meeker* and *Escobar* [22.5]. Often these tests must be accelerated to obtain timely information.

ATs can be characterized by the nature of the response variable in the test (i. e., what can be measured or observed, relative to reliability):

- **Accelerated Life Tests ALT**
  The response in an ALT is related to the lifetime of the product. Often ALT data are right-censored because the test is stopped before all units fail. In other cases, the ALT response is interval-censored because failures are discovered at particular inspection times.
- **Accelerated Repeated Measures Degradation Tests ARMDT**

In an ARMDT, one measures degradation on a sample of units at different points in time. In general, each unit provides several degradation measurements. The degradation response could be actual chemical or physical degradation or performance degradation (e.g., drop in power output).

- **Accelerated Destructive Degradation Tests ADDT**
  An ADDT is similar to an ARMDT, except that the measurements are destructive, so one can obtain only one observation per test unit.

These different kinds of ATs can be closely related because they can involve the same underlying physical/chemical mechanisms for failure and models for acceleration. They are different, however, in that different kinds of statistical models and analyses are performed because of the differences in the kind of response.

This chapter focuses on analyses of data from ALTs. See *Meeker* and *Escobar* ([22.3], Chapts. 13 and 21) for further discussion of ARMDTs and see *Nelson* ([22.1], Chapt. 11) and *Meeker*, *Escobar*, *Kugler*, and *Kramer* [22.6] for models, methods, and examples pertaining to ADDTs.

22.1.2 Methods of Acceleration

There are different methods of accelerating tests to induce product failures more quickly. These methods vary depending on the nature of the product or material being tested.

- **Accelerate the Product Use Rate**
  This method is appropriate for products that are ordinarily not in continuous use. For example, the median life of a bearing for a certain washing machine agitator is 12 years, based on an assumed use rate of eight loads per week. If the machine is tested at 112 loads per week (16 per day), the median life is reduced to roughly 10 months.
- **Accelerate Product Aging**
  Changing environmental conditions (e.g., increasing humidity or temperature) can be used to increase the rate of chemical degradation of products such as insulating materials and adhesive bonds.
- **Accelerate by Increasing Product Stress**
  Increasing stress (e.g., voltage or pressure) on a specimen will generally cause it to fail sooner.

It is also possible to accelerate product failures by using combinations of these accelerating variables. For example, in fatigue testing one uses higher cycling rates and higher than usual levels of stress. In electro-chemical reactions, increasing voltage will also increase the rate of chemical change. Putting higher voltage stress on an insulating material may generate heat that will accelerate chemical change. In all types of acceleration, care should be taken to make sure that the underlying mechanisms and the resulting failure modes in an AT are the same as those that will affect the product in actual use.

22.1.3 Choosing an Accelerated Life Test Model

The task of finding an ALT model can be divided into two steps:

1. Choose an appropriate statistical distribution to describe lifetime at fixed levels of the accelerating variable(s). Typically the same distribution is used at all levels of stress, as would be suggested by the commonly used scale-accelerated failure-time, SAFT, model. Probability plots (i. e., plotting a nonparametric estimate on special distribution-specific probability scales) are used to help identify an appropriate distribution.
2. Choose a model to describe the relationship between the lifetime distributions and the accelerating variables. It is best if the selected model is based on physical or chemical theory, empirical considerations, or both.

These two steps are discussed in detail in Sects. 22.2 and 22.3, respectively.

As will be discussed in subsequent sections, the combination of probability plotting and maximum likelihood provides the basic statistical tools for analyzing most ALT data. Probability plots provide a convenient, intuitive method to display ALT data and assess agreement with proposed models for the data. These methods are described in detail in *Nelson* ([22.1], Chapt. 3) and *Meeker* and *Escobar* ([22.3], Chapts. 6 and 19). Maximum likelihood (ML) is the most common method of fitting an ALT model to data because it allows for censored data and because ML estimates have desirable statistical properties. The ML estimates of model parameters are those parameter values that maximize the probability of the observed data for the proposed model. The theory of ML methods has been thoroughly explored in the literature. *Nelson* ([22.1], Chapt. 5) and *Meeker* and *Escobar* ([22.3], Chapts. 8 and 19) discuss ML methods relevant to ALT data analysis. Both probability plotting and maximum likelihood methods are now widely available in commercial statistical software, as described in Sect. 22.9.

22.2 Life Distributions

In this chapter, the symbol T will denote lifetime for devices, components, systems, etc. Then, it will be assumed that T is a positive continuous random variable. The cumulative distribution function CDF for T is the probability of failure by time t (or the fraction failing by time t) and one writes $F(t) = \Pr(T \le t)$. The probability density function PDF $f(t)$ is the derivative of the CDF. That is, $f(t) = \mathrm{d}F(t)/\mathrm{d}t$. Selecting a probability distribution for lifetime is equivalent to specifying either $F(t)$ or $f(t)$.

This section describes log-location-scale probability distributions. Important members of this family include the popular Weibull and lognormal distributions. Other members of the family are described in Chapt. 4 of *Meeker* and *Escobar* [22.3]. Distributions that are other than log-location-scale (which could also be used in ALT modeling) are described in Chapt. 5 of *Meeker* and *Escobar* [22.3].

A random variable Y has a location-scale distribution if its CDF can be written as

$$F(y;\mu,\sigma) = \Pr(Y \le y) = \Phi\left(\frac{y-\mu}{\sigma}\right) ,$$

where μ is a location parameter, σ is a scale parameter, and Φ does not depend on any unknown parameters. In many reliability applications, it is assumed that $\log(T)$ has a location-scale distribution. Then T is said to have a log-location-scale distribution.

22.2.1 The Lognormal Distribution

Lifetime T has a lognormal distribution if its CDF and PDF are

$$F(t;\mu,\sigma) = \Phi_{\mathrm{nor}}\left(\frac{\log(t)-\mu}{\sigma}\right) ,$$

$$f(t;\mu,\sigma) = \frac{1}{\sigma t}\phi_{\mathrm{nor}}\left(\frac{\log(t)-\mu}{\sigma}\right) , \quad t > 0 ,$$

where Φ_{nor} and ϕ_{nor} are the standard normal (Gaussian) CDF and PDF, respectively. In particular,

$$\phi_{\mathrm{nor}}(z) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2}{2}\right) .$$

The parameters (μ,σ) are the mean and the standard deviation of $\log(T)$, respectively. Then $[\exp(\mu),\sigma]$ are, respectively, the scale and shape parameters of T. The lognormal p quantile is $t_p = \exp[\mu + \Phi^{-1}_{\mathrm{nor}}(p)\,\sigma]$. Note that $\exp(\mu)$ corresponds to the median lifetime. That is, $t_{0.50} = \exp(\mu)$.

22.2.2 The Weibull Distribution

Lifetime T has a Weibull distribution if its CDF and PDF are

$$\begin{aligned} F(t;\mu,\sigma) &= \Phi_{\mathrm{sev}}\left(\frac{\log(t)-\mu}{\sigma}\right) \\ &= 1-\exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \\ f(t;\mu,\sigma) &= \frac{1}{\sigma t}\phi_{\mathrm{sev}}\left(\frac{\log(t)-\mu}{\sigma}\right) \\ &= \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}\exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right], \quad t > 0, \end{aligned}$$

where Φ_{sev} and ϕ_{sev} are the standard smallest extreme value SEV CDF and PDF defined by

$$\Phi_{\mathrm{sev}}(z) = 1-\exp[-\exp(z)]$$
$$\text{and} \quad \phi_{\mathrm{sev}}(z) = \exp[z-\exp(z)] ,$$

$\sigma > 0$, and $-\infty < \mu < \infty$. Here (μ,σ) are the location and scale parameters for the distribution of $\log(T)$. The expressions for the CDF and PDF above also use $\eta = \exp(\mu)$ and $\beta = 1/\sigma$, the traditional Weibull scale and shape parameters, respectively. The Weibull p quantile is $t_p = \exp\{\mu + \log[-\log(1-p)]\,\sigma\} = \eta[-\log(1-p)]^{\beta}$. Note that η is approximately the 0.63 quantile of the Weibull distribution.

The parameterization in terms of (μ,σ) is particularly convenient for lifetime regression models and is used extensively in this chapter. The (η,β) parameterization is commonly used in the engineering and statistical literature.

22.3 Acceleration Models

fitting a model to data obtained at high levels of the accelerating variables and extrapolating the results to use conditions levels. Ideally, this model should be derived from physical or chemical theory and verified empirically to justify the extrapolation. If a physical understanding of the failure mechanism is lacking, an

empirical model might be justified for extrapolation if it is based on extensive experience with how failure mechanisms relate to accelerating variables.

This section discusses basic ideas of acceleration models and some physical considerations that lead to these models. For more information on these models, see *Nelson* ([22.1], Chapt. 2) and *Meeker* and *Escobar* ([22.3], Chapt. 18).

22.3.1 Scale-Accelerated Lifetime Model

A simple, commonly used model used to characterize the effect that explanatory variables $\boldsymbol{x} = (x_1, \ldots, x_k)'$ have on lifetime T is the scale-accelerated failure-time (SAFT) model. Some of these explanatory variables are accelerating, but others may just be of interest (e.g., for product design optimization decisions). If $\boldsymbol{x}_\mathrm{U}$ denotes the ordinary use conditions, under a SAFT model, lifetime at $\boldsymbol{x}$, $T(\boldsymbol{x})$, is scaled by a deterministic factor that might depend on $\boldsymbol{x}$ and unknown fixed parameters. More specifically, $T(\boldsymbol{x}) = T(\boldsymbol{x}_\mathrm{U})/\mathcal{AF}(\boldsymbol{x})$ where the *acceleration factor* $\mathcal{AF}(\boldsymbol{x})$ is a positive function of $\boldsymbol{x}$ satisfying $\mathcal{AF}(\boldsymbol{x}_\mathrm{U}) = 1$. Lifetime is accelerated (decelerated) when $\mathcal{AF}(\boldsymbol{x}) > 1$ [$\mathcal{AF}(\boldsymbol{x}) < 1$]. Some special cases of these important SAFT models are discussed in the following sections.

Observe that under a SAFT model, the probability that failure under conditions $\boldsymbol{x}$ occurs at or before time t can be written as $\Pr[T(\boldsymbol{x}) \le t] = \Pr[T(\boldsymbol{x}_\mathrm{U}) \le \mathcal{AF}(\boldsymbol{x}) \times t]$. As described in Sect. 22.2, it is common practice to assume that the lifetime $T(\boldsymbol{x})$ has a log-location-scale distribution such as a lognormal or Weibull distribution in which μ is a function of the accelerating variable(s) and σ is constant (i. e., does not depend on $\boldsymbol{x}$). In this case,

$$F(t; \boldsymbol{x}_\mathrm{U}) = \Pr[T(\boldsymbol{x}_\mathrm{U}) \le t] = \Phi\left(\frac{\log(t) - \mu_\mathrm{U}}{\sigma}\right), \tag{22.1}$$

where Φ denotes a standard cumulative distribution function (e.g., standard normal or standard smallest extreme value) and μ_U is the location parameter for the distribution of $\log[T(\boldsymbol{x}_\mathrm{U})]$. Thus,

$$F(t; \boldsymbol{x}) = \Pr[T(\boldsymbol{x}) \le t] = \Phi\left(\frac{\log(t) - \{\mu_\mathrm{U} - \log[\mathcal{AF}(\boldsymbol{x})]\}}{\sigma}\right). \tag{22.2}$$

Note that $T(\boldsymbol{x})$ also has a log-location-scale distribution with location parameter $\mu = \mu_\mathrm{U} - \log[\mathcal{AF}(\boldsymbol{x})]$ and a scale parameter σ that does not depend on $\boldsymbol{x}$.

22.3.2 Accelerating Product Use Rate

Increasing the use rate can be an effective method of acceleration for some products. In simple situations the cycles-to-failure distribution does not depend on the cycling rate. In such situations *reciprocity holds*. Then the underlying model for lifetime is SAFT where $\mathcal{AF}(\mathrm{UseRate}) = \mathrm{UseRate}/\mathrm{UseRate}_\mathrm{U}$ is the factor by which the test is accelerated.

Use-rate acceleration may be appropriate for products such as electrical relays and switches, paper copiers, and printers, and home appliances such as toasters and washing machines. The manner in which the use rate is increased may depend on the product. For example, *Nelson* ([22.1], page 16) states that failure of rolling bearings can be accelerated by running them at three or more times the normal speed. *Johnston* et al. [22.7] demonstrated that the cycles-to-failure of electrical insulation was shortened, approximately, by a factor of $\mathcal{AF}(412) = 412/60 \approx 6.87$ when the applied AC voltage in endurance tests was increased from 60 Hz to 412 Hz.

ALTs with increased use rate attempt to simulate actual use. Thus other environmental factors should be controlled to mimic actual use environments. If the cycling rate is too high, it can cause *reciprocity breakdown*. For example it may be necessary to have test units (such as a toaster) cool down between cycles of operation. *Dowling* ([22.8], page 706) describes how increased cycle rate may affect the crack growth rate in per-cycle fatigue testing.

Reciprocity breakdown is known to occur, for example, for certain components in copying machines where components tend to last longer (in terms of cycles) when printing is done at higher rates. In such cases, the empirical power-rule relationship $\mathcal{AF}(\mathrm{UseRate}) = (\mathrm{UseRate}/\mathrm{UseRate}_\mathrm{U})^p$ is often used, where p can be estimated by testing at two or more use rates.

22.3.3 Models for Temperature Acceleration

This section describes common models that are used to describe the relationship between lifetime and temperature.

The Arrhenius Relationship for Temperature Acceleration

The Arrhenius equation is widely used to relate the rate of a chemical reaction $\mathcal{R}$ to temperature temp. This relationship can be written as

$$\mathcal{R}(\text{temp}) = \gamma_0 \exp\left(\frac{-E_a}{k_B \times \text{temp K}}\right) = \gamma_0 \exp\left(\frac{-E_a \times 11605}{\text{temp K}}\right), \quad (22.3)$$

where γ_0 and the activation energy E_a are constants that depend on material properties and test methods, $k_B = 8.6171 \times 10^{-5} = 1/11605$ is Boltzmann's constant in units of electron volts per °C, and temp K = temp °C + 273.15 is the temperature kelvin. The Arrhenius lifetime relationship is based on the view that failure occurs after there has been a critical amount of chemical reaction. Then when $\mathcal{R}$(temp) is larger, the failure will occur sooner (e.g., *Klinger* [22.9]). Empirical observations have suggested that the Arrhenius equation provides an adequate description of the relationship between product life and temperature in a variety of applications such as integrated circuits (several different kinds of failure modes), light-emitting diodes (LEDs), adhesive bonds, lubricants, incandescent bulb filaments, insulating tapes, and dielectric materials. It should be noted, however, that the nature of the failure mechanism may limit the range of temperature over which the Arrhenius relationship is adequate.

Let temp_U be the temperature at use conditions. Then the Arrhenius acceleration factor is

$$\mathcal{AF}(\text{temp}, \text{temp}_U, E_a) = \frac{\mathcal{R}(\text{temp})}{\mathcal{R}(\text{temp}_U)} = \exp\left[E_a\left(\frac{11605}{\text{temp}_U\,\text{K}} - \frac{11605}{\text{temp K}}\right)\right].$$

For simplicity, the time-acceleration factor is sometimes written as $\mathcal{AF}$(temp) instead of $\mathcal{AF}(\text{temp}, \text{temp}_U, E_a)$.

Let $\text{temp}_{Low} < \text{temp}_{High}$ be two temperature levels and define the temperature differential factor TDF

$$\text{TDF} = \left(\frac{11605}{\text{temp}_{Low}\,\text{K}} - \frac{11605}{\text{temp}_{High}\,\text{K}}\right). \quad (22.4)$$

Then, one can write

$$\mathcal{AF}(\text{temp}_{High}, \text{temp}_{Low}, E_a) = \exp(E_a \times \text{TDF}). \quad (22.5)$$

Example 22.1: Adhesive-Bonded Power Element. *Meeker* and *Hahn* [22.10] describe an adhesive-bonded power element that was designed for use at temp = 50 °C and a life test of this element is to be conducted at temp = 120 °C. Suppose that experience with this product suggested that $E_a = 0.5$. Using (22.4) and (22.5) gives TDF = 6.39 and then $\mathcal{AF}(120) \approx 24.41$ gives the acceleration factor for the chemical reaction when testing the power element at 120 °C.

Note that the Arrhenius relationship (and the assumption that the failure is directly related to the amount of material reacting) implies that the acceleration model is a SAFT model. If $T(\text{temp}_U)$ has a log-location-scale distribution with parameters μ_U and σ, then it follows from (22.2) that T(temp) also has a log-location-scale distribution with parameters $\mu = \mu_U - \beta_1 x$ and σ, where $\beta_1 = E_a$, $x = \text{TDF}$, and the TDF is computed for the temperatures $\text{temp}_U < \text{temp}$.

Nonlinear Degradation Path Models for Reaction-Rate Acceleration

As discussed earlier, when failure is caused directly by degradation, one can relate the distribution of degradation to the distribution of failure. Suppose that failure occurs when degradation reaches a certain critical level $\mathcal{D}_f$. This section discusses the situation in which degradation follows a nonlinear path (e.g., the degradation path will reach an asymptote because of a limited amount of material that is available to go into the reaction) over time. The linear path case is discussed in the next section.

The amount of degradation at time t and temperature temp is given by

$$\mathcal{D}(t; \text{temp}) = \mathcal{D}_\infty \times \left\{1 - \exp\left[-\mathcal{R}_U \times \mathcal{AF}(\text{temp}) \times t\right]\right\}, \quad (22.6)$$

where $\mathcal{R}_U$ is the rate of reaction rate at the use temperature temp_U. Note that $\mathcal{R}_U \times \mathcal{AF}$(temp) is the rate reaction at temp and $\mathcal{AF}(\text{temp}) > 1$ when $\text{temp} > \text{temp}_U$. Figure 22.1 shows the decrease in strength of adhesive bonds, as a function of time for different temperatures. In this application, $\mathcal{D}_\infty < 0$ so that, for fixed temp, $\mathcal{D}(t; \text{temp})$ is a decreasing function of t and failure occurs when $\mathcal{D}(t; \text{temp})$ falls below $\mathcal{D}_f$, say 50 N. In other applications, $\mathcal{D}_\infty > 0$, then $\mathcal{D}(t; \text{temp})$ is increasing and failure occurs once $\mathcal{D}(t; \text{temp})$ exceeds $\mathcal{D}_f$. For both cases, the lifetime T(temp) at any level temp is given by $T(\text{temp}) = T(\text{temp}_U)/\mathcal{AF}(\text{temp})$ where $T(\text{temp}_U) = -(1/\mathcal{R}_U)\log(1 - \mathcal{D}_f/\mathcal{D}_\infty)$. Note that this has the form of a SAFT model. Sufficient

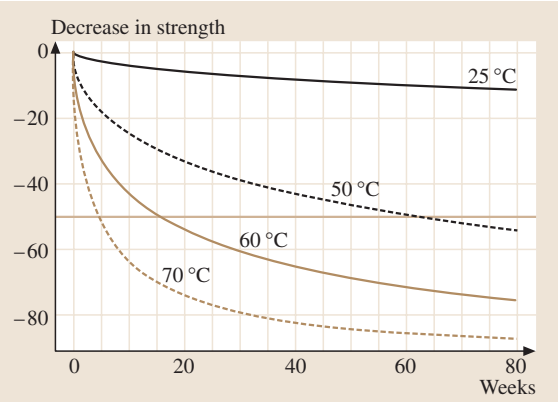


Fig. 22.1 Nonlinear decreasing degradation path at different temperatures for a SAFT model

conditions for a degradation model to be SAFT are given by *LuValle*, *Welsher* and *Svoboda* [22.11] and *Klinger* [22.9]. Under a SAFT model and a log-location-scale distribution with parameters μ and σ for $T(\text{temp}_U)$, $T(\text{temp})$ at any temp also has a log-location-scale distribution with the same σ and

$$\mu = \mu_U - \log\left[\mathcal{AF}(\text{temp})\right] = \beta_0 + \beta_1 x,$$

where $x = 11605/(\text{tempK})$, $x_U = 11605/(\text{temp}_U\text{K})$, $\beta_1 = E_a$, $\beta_0 = \mu_U - \beta_1 x_U$, and μ_U is the location parameter of the distribution of $\log(T)$ at use conditions.

Linear Degradation Path Models for Reaction-Rate Acceleration

There are situations in which a linear model can be used to approximate degradation over time. For example, when $\mathcal{R}_U \times \mathcal{AF}(\text{temp}) \times t$ in (22.6) is small enough so that $\mathcal{D}(t)$ is small compared to $\mathcal{D}_\infty$, then

$$\begin{aligned}\mathcal{D}(t;\text{temp}) &= \mathcal{D}_\infty \times \left\{1 - \exp\left[-\mathcal{R}_U \times \mathcal{AF}(\text{temp}) \times t\right]\right\} \\ &\approx \mathcal{D}_\infty \times \mathcal{R}_U \times \mathcal{AF}(\text{temp}) \times t = \mathcal{R}_U^+ \times \mathcal{AF}(\text{temp}) \times t\,,\end{aligned}$$

where $\mathcal{R}_U^+ = \mathcal{D}_\infty \times \mathcal{R}_U$. There are also practical situations (e.g., wear of automobile tires) for which a linear path adequately approximates degradation over time. In this case, $\mathcal{D}(t;\text{temp}) = \mathcal{R}_U^+ \times \mathcal{AF}(\text{temp}) \times t$ where it is assumed that $\mathcal{D}(0;\text{temp}) = 0$ and that $\mathcal{R}_U^+ \times \mathcal{AF}(\text{temp})$ is the degradation rate at condition temp.

Again, failure occurs when $\mathcal{D}(t;\text{temp})$ reaches a critical level $\mathcal{D}_f$. The equation $\mathcal{D}(t;\text{temp}) = \mathcal{D}_f$ yields the lifetime at temp, $T(\text{temp}) = T(\text{temp}_U)/\mathcal{AF}(\text{temp})$ which, again, is in SAFT form. Thus, if $T(\text{temp})$ has a log-location-scale distribution, $\mu = \beta_0 + \beta_1 x$ and σ does not depend on x (where x has the same definition given in Sect. 22.3.3).

22.3.4 Models for Voltage and Voltage–Stress Acceleration

Voltage or voltage stress can also be used to accelerate degradation and hasten product failures. Voltage measures the amount of force needed to move an electric charge between two points. Such a flow of charges produces an electrical current. Voltage stress measures voltage per unit of thickness of a dielectric. For dielectric components (e.g., capacitors and insulators), chemical degradation reduces the dielectric strength over time. Also, stronger electric fields can accelerate the growth of discontinuities, electrochemical reactions, or electromechanical changes that cause failure.

Example 22.2: Accelerated Life Test of Insulation for Generator Armature Bars. *Doganaksoy* et al. [22.12] discuss an ALT for a new mica-based insulation design for generator armature bars GAB. Degradation of an organic binder in the insulation causes a decrease in voltage strength and this was the primary cause of failure in the insulation. The insulation was designed for use at a voltage stress of 120 V/mm. Voltage-endurance tests were conducted on 15 electrodes at each of five accelerated voltage levels between 170 V/mm and 220 V/mm (i. e., a total of 75 electrodes). Each test was run for 6480 h at which point 39 of the electrodes had not yet failed. Table 22.1 gives the data from these tests. The insulation engineers were interested in the 0.01 and 0.05 quantiles of lifetime at the use condition of

Table 22.1 GAB insulation data

Voltage stress (V/mm)	Lifetime (10^3 h)
170	15 censored[a]
190	3.248, 4.052, 5.304, 12 censored[a]
200	1.759, 3.645, 3.706, 3.726, 3.990, 5.153, 6.368, 8 censored[a]
210	1.401, 2.829, 2.941, 2.991, 3.311, 3.364, 3.474, 4.902, 5.639, 6.021, 6.456, 4 censored[a]
220	0.401, 1.297, 1.342, 1.999, 2.075, 2.196, 2.885, 3.019, 3.550, 3.566, 3.610, 3.659, 3.687, 4.152, 5.572

[a] Units were censored at 6.480 10^3 h

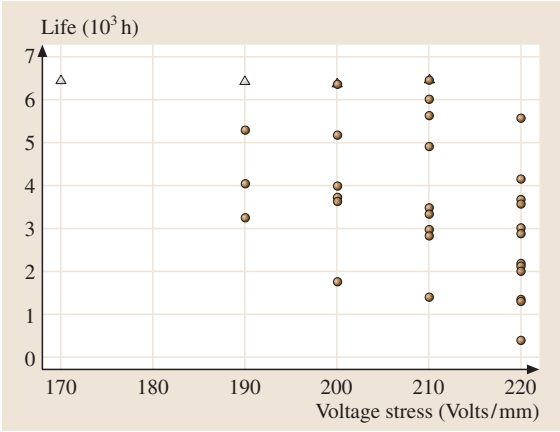


Fig. 22.2 GAB insulation data. Scatter plot of life versus voltage. Censored observations are indicated by Δ

120 V/mm. Figure 22.2 plots the insulation lifetimes against voltage stress.

Inverse Power Relationship

The inverse power relationship is frequently used to describe the effect that stresses like voltage and pressure have on lifetime. Voltage is used in the following discussion. When the thickness of a dielectric material or insulation is constant, voltage is proportional to voltage stress. Let volt denote voltage and let volt_U be the voltage at use conditions. The lifetime at stress level volt is given by

$$T(\text{volt}) = \frac{T(\text{volt}_\text{U})}{\mathcal{AF}(\text{volt})} = \left(\frac{\text{volt}}{\text{volt}_\text{U}}\right)^{\beta_1} T(\text{volt}_\text{U}) \ ,$$

where β_1, in general, is negative. The model has SAFT form with acceleration factor

$$\mathcal{AF}(\text{volt}) = \mathcal{AF}(\text{volt}, \text{volt}_\text{U}, \beta_1) = \frac{T(\text{volt}_\text{U})}{T(\text{volt})} = \left(\frac{\text{volt}}{\text{volt}_\text{U}}\right)^{-\beta_1} . \quad (22.7)$$

If $T(\text{volt}_\text{U})$ has a log-location-scale distribution with parameters μ_U and σ, then $T(\text{volt})$ also has a log-location-scale distribution with $\mu = \beta_0 + \beta_1 x$, where $x_\text{U} = \log(\text{volt}_\text{U})$, $x = \log(\text{volt})$, $\beta_0 = \mu_\text{U} - \beta_1 x_\text{U}$, and σ does not depend on x.

Example 22.3: Time Acceleration for GAB Insulation. For the GAB insulation data in Example 22.2, an estimate for β_1 is $\hat{\beta}_1 = -9$ (methods for computing such estimates are described and illustrated in Sect. 22.4). Recall that the design voltage stress is $\text{volt}_\text{U} = 120$ V/mm and consider testing at volt = 210 V/mm. Thus, using $\beta_1 = \hat{\beta}_1$, $\mathcal{AF}(210) = (210/120)^9 \approx 154$. Thus by increasing voltage stress from 120 to 210 V/mm, one estimates that lifetime is shortened by a factor of $1/\mathcal{AF}(210) \approx 1/154 = 0.0065$. Figure 22.3 plots $\mathcal{AF}$ versus volt for $\beta_1 = -7, -9, -11$. Using direct computations or from the plot, one obtains $\mathcal{AF}(210) \approx 50$ for $\beta_1 = -7$ and $\mathcal{AF}(210) \approx 471$ for $\beta_1 = -11$.

Motivation for the Inverse Power Relationship

The following description of a failure process that relates lifetime to pressure-like stresses, leading to the inverse power relationship, comes from *Meeker* and *Escobar* ([22.3], Sect. 18.4.3). Insulation units or specimens will have a characteristic dielectric strength $\mathcal{D}$. This property varies from unit to unit and degrades over time. When $\mathcal{D}$ degrades to the level of the applied voltage stress volt, a failure-causing event (e.g., short circuit or a flash-over) is triggered. Suppose one can express the dielectric strength at time t by $\mathcal{D}(t) = \delta_0 \times t^{1/\beta_1}$. Letting $\mathcal{D}(t) = \text{volt}$ and solving for t, $T(\text{volt}) = (\text{volt}/\delta_0)^{\beta_1}$. The acceleration factor for comparing lifetimes at volt and volt_U is

$$\mathcal{AF}(\text{volt}) = \mathcal{AF}(\text{volt}, \text{volt}_\text{U}, \beta_1) = \frac{T(\text{volt}_\text{U})}{T(\text{volt})} = \left(\frac{\text{volt}}{\text{volt}_\text{U}}\right)^{-\beta_1} ,$$

which agrees with (22.7).

The inverse power relationship can be used to describe cases for which the accelerating voltage increases the degradation rate. For example, suppose that degra-

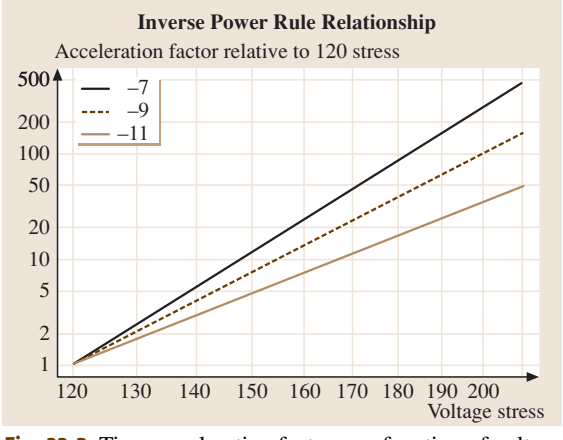


Fig. 22.3 Time-acceleration factor as a function of voltage stress and exponent $-\beta_1 = -7, -9, -11$

dation is more appropriately described by

$$\mathcal{D}(t) = \delta_0 \left[\mathcal{R}(\text{volt}) \times t\right]^{1/\gamma_1} ,$$

where $\mathcal{R}(\text{volt}) = \gamma_0 \exp\left[\gamma_2 \log(\text{volt})\right]$. Again, if failure occurs when $\mathcal{D}_\text{f} = \text{volt}$, then lifetime is

$$T(\text{volt}) = \frac{1}{\mathcal{R}(\text{volt})} \left(\frac{\text{volt}}{\delta_0}\right)^{\gamma_1} .$$

Then the acceleration factor is

$$\mathcal{AF}(\text{volt}) = \frac{T(\text{volt}_\text{U})}{T(\text{volt})} = \left(\frac{\text{volt}}{\text{volt}_\text{U}}\right)^{\gamma_2 - \gamma_1}$$

which is an inverse power relationship with $\beta_1 = \gamma_1 - \gamma_2$.

This discussion can be extended to other materials and products such as filaments of light bulbs, capacitors, roller bearings, and ball bearings. The inverse power relationship has also been used with other accelerating factors such as loading stress, cycling rate, humidity, and pressure.

22.3.5 Models for Two-or-More-Variable Acceleration

Some ALTs use two or more accelerating variables that affect the degradation process. In comparison with single-variable ALTs, in some situations, the degradation rate can be increased more effectively when two or more variables are used to accelerate the test. Many ALTs use temperature in combination with another environmental condition such as pressure, voltage, and humidity. The rest of this section describes models for these ALTs.

Extending the Arrhenius Relationship

Recall the Arrhenius equation (22.3), which gives the rate of a chemical reaction as a function of temperature. Suppose that X represents a non-thermal accelerating variable. A model for reaction rate in this situation is

$$\mathcal{R}(\text{temp}, X) = \gamma_0 \times \exp\left(\frac{-\gamma_1}{k_\text{B} \times \text{temp K}}\right) \times \exp\left(\gamma_2 X + \frac{\gamma_3 X}{k_\text{B} \times \text{temp K}}\right) , \quad (22.8)$$

where the parameters $\gamma_1 = E_\text{a}$ (the effective activation energy), γ_0, γ_2, and γ_3 are specific properties of the failure-causing chemical reaction. In (22.8), the ratio $\gamma_3 X/(k_\text{B} \times \text{temp K})$ represents possible interaction between the two accelerating variables. This model could be further extended by adding appropriate factors for other explanatory variables to the right-hand side of (22.8).

Using a SAFT formulation, one can equivalently express the extended Arrhenius relationship (22.8) in terms of the acceleration factor given by

$$\mathcal{AF}(\text{temp}, X) = \frac{\mathcal{R}(\text{temp}, X)}{\mathcal{R}(\text{temp}_\text{U}, X_\text{U})} . \quad (22.9)$$

When $T(\text{temp}_\text{U}, X_\text{U})$ has a log-location-scale distribution, (22.1), (22.8) and (22.9) imply that $T(\text{temp}, X)$ also has a log-location-scale distribution with

$$\begin{aligned}\mu &= \mu_\text{U} - \log[\mathcal{AF}(\text{temp}, X)] \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 ,\end{aligned} \quad (22.10)$$

where $\beta_1 = E_\text{a}$, $\beta_2 = -\gamma_2$, $\beta_3 = -\gamma_3$, $x_1 = 11605/(\text{temp K})$, $x_2 = X$, $\beta_0 = \mu_\text{U} - \beta_1 x_{1\text{U}} - \beta_2 x_{2\text{U}} - \beta_3 x_{1\text{U}} x_{2\text{U}}$, and σ does not depend on the explanatory variables (temp, X).

Temperature–Voltage Acceleration Models

There has been a variety of approaches that have been used to model the combination of temperature and voltage acceleration. For instance, *Meeker* and *Escobar* ([22.3], Sect. 17.7) analyzed data from a study relating voltage and temperature to the failure of glass capacitors. They modeled the location parameter of log lifetime as a simple linear function of temp °C and volt. The extended Arrhenius relationship in Sect. 22.3.5 can also be used with $X = \log(\text{volt})$, as in *Boyko* and *Gerlach* [22.13]. *Klinger* [22.14] modeled the *Boyko* and *Gerlach* [22.13] data by including second-order terms for both accelerating variables.

To derive the time-acceleration factor for the extended Arrhenius relationship with temp and volt, one can follow steps analogous to those outlined in Sect. 22.3.4. Using the dielectric strength (degradation path) model at time t, $\mathcal{D}(t) = \delta_0 \left[\mathcal{R}(\text{temp}, \text{volt}) \times t\right]^{1/\gamma_1}$. Using (22.8) with $X = \log(\text{volt})$, we obtain

$$\mathcal{R}(\text{temp}, \text{volt}) = \gamma_0 \times \exp\left(\frac{-E_\text{a}}{k_\text{B} \times \text{temp K}}\right) \times \exp\left[\gamma_2 \log(\text{volt}) + \frac{\gamma_3 \log(\text{volt})}{k_\text{B} \times \text{temp K}}\right] .$$

Again, failure occurs when the dielectric strength crosses the applied voltage stress, that is, $\mathcal{D}(t) = \text{volt}$. This occurs at time

$$T(\text{temp}, \text{volt}) = \frac{1}{\mathcal{R}(\text{temp}, \text{volt})} \left(\frac{\text{volt}}{\delta_0}\right)^{\gamma_1} .$$

From this, one computes

$$\mathcal{AF}(\text{temp, volt}) = \frac{T(\text{temp}_U, \text{volt}_U)}{T(\text{temp, volt})} = \exp[E_a(x_{1U} - x_1)] \times \left(\frac{\text{volt}}{\text{volt}_U}\right)^{\gamma_2-\gamma_1} \times \{\exp[x_1 \log(\text{volt}) - x_{1U} \log(\text{volt}_U)]\}^{\gamma_3}.$$

where $x_{1U} = 11605/(\text{temp}_U\,\text{K})$ and $x_1 = 11605/(\text{temp K})$. When $\gamma_3 = 0$, there is no interaction between temperature and voltage. In this case, $\mathcal{AF}(\text{temp, volt})$ can be factored into two terms, one that involves temperature only and another term that involves voltage only. Thus, if there is no interaction, the contribution of temperature (voltage) to acceleration is the same at all levels of voltage (levels of temperature).

Temperature–Current Density Acceleration Models

d'Heurle and *Ho* [22.15] and *Ghate* [22.16] studied the effect of increased current density (A/cm^2) on electromigration in microelectronic aluminum conductors. High current densities cause atoms to move more rapidly, eventually causing extrusion or voids that lead to component failure. ATs for electromigration often use increased current density and temperature to accelerate the test. An extended Arrhenius relationship could be appropriate for such data. In particular, when T has a log-location-scale distribution, then (22.10) applies with $x_1 = 11605/\text{temp K}$, $x_2 = \log(\text{current})$. The model with $\beta_3 = 0$ (without interaction) is known as Black's equation (*Black* [22.17]).

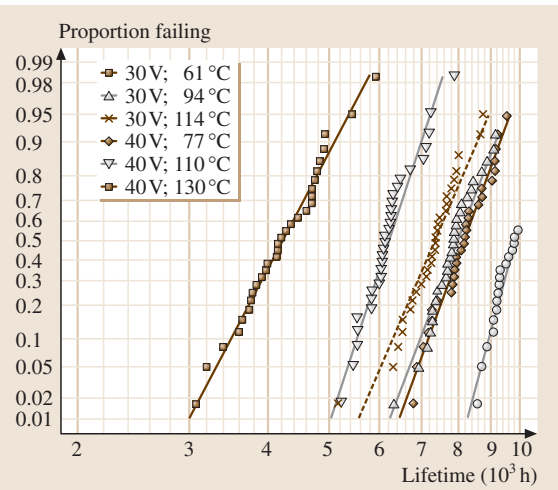


Fig. 22.4 LED device data. Lognormal multiple probability plot with lognormal ML fits for each combination of voltage/temperature

Example 22.4: Light Emitting Device (LED) Example. A degradation study on a light emitting device LED was conducted to study the effect of current and temperature on light output. A unit was said to have failed if its output was reduced to 60% of its initial value. Two levels of current and six levels of temperature were used in the test. Figure 22.4 is a probability plot of lifetimes for each combination of current and temperature. The plot suggests that the lognormal distribution is an appropriate distribution for lifetime and that increasing either temperature or current results in shorter device life.

Temperature–Humidity Acceleration Models

Relative humidity is another environmental variable that can be combined with temperature to accelerate corrosion or other chemical reactions. Examples of applications include paints and coatings, electronic devices and electronic semiconductor parts, circuit boards, permalloy specimens, foods, and pharmaceuticals. Although most ALT models that include humidity were derived empirically, some humidity models have a physical basis. For example, *Gillen* and *Mead* [22.18] and *Klinger* [22.19] studied kinetic models relating aging with humidity. *LuValle* et al. [22.20] provided a physical basis for studying the effect of humidity, temperature, and voltage on the failure of circuit boards. See *Boccaletti* et al. [22.21], *Nelson* ([22.1], Chapt. 2) *Joyce* et al. [22.22], *Peck* [22.23], and *Peck* and *Zierdt* [22.24] for ALT applications involving temperature and humidity.

The extended Arrhenius relationship (22.10)) applied to ALTs with temperature and humidity uses $x_1 = 11605/\text{temp K}$, $x_2 = \log(\text{RH})$, and $x_3 = x_1 x_2$, where RH is a proportion denoting relative humidity. The case when $\beta_3 = 0$ (no temperature–humidity interaction) is known as Peck's relationship and was used by Peck [22.23] to study failures of epoxy packing. Klinger [22.19] suggested the term $x_2 = \log[\text{RH}/(1 - \text{RH})]$ instead of $\log(\text{RH})$. This alternative relationship is based on a kinetic model for corrosion.

The Eyring Model

Fundamental work on the relationship between reaction rates and temperature, in combination with other variables, was done by Eyring in a number of important works (*Eyring*, *Gladstones*, and *Laidler* [22.25] and *Eyring* [22.26]). The Eyring model is the same as the extended Arrhenius model described in this section, except

that it has an additional factor $g(\text{temp K})$ outside of the exponential in (22.8). Usually, $g(\text{temp K}) = (\text{temp K})^p$ is used. For the purposes of using a fitted acceleration model to predict life at use conditions, for any practical values of the model parameters, this extra factor and estimating p along with E_a have no perceptible effect on the prediction. Thus the simpler Arrhenius model is what is generally used in practice. See *Meeker* and *Escobar* ([22.3], Sect. 18.3.2) and the above references for more information on the Eyring model.

22.3.6 Guidelines and Issues for Using Acceleration Models

The main objective of AT is to collect data at high levels of accelerating variables and extrapolate the results to use conditions. This section provides some guidelines and lists some common issues that arise in fitting acceleration models to ALT data.

- Whenever possible, the acceleration model should have a basis in physical or chemical theory. A careful review of previous theoretical and empirical work will often shed light on which environmental factors and degradation processes are relevant to a failure mechanism. It is highly recommended that a group of experts oversee the entire project of planning, collecting and analyzing AT data. This group should include experts with a clear understanding of the mechanical, physical and chemical nature of how the product degrades and eventually fails. This team should also include someone who is knowledgeable of the statistical aspects of planning life tests and analyzing the resulting data.
- ATs often assume a relatively simple failure mechanism or degradation process. It is possible at high levels of an accelerating variable to introduce new failure mechanisms. Acceleration could also alter other environmental factors that affect the failure mechanism. For example, *Meeker* and *Escobar* ([22.3], page 526) pointed out that ALT tests of circuit packs at higher temperatures reduced humidity, causing fewer failures than expected. Potential scenarios like these should be considered for both planning and modeling ATs.
- The main difficulty with extrapolation in ATs is that there is rarely sufficient data at lower levels of the accelerating variable to verify that the acceleration model for the failure mechanism still holds at use conditions. Thus, a good test plan minimizes the amount of acceleration while providing a statistically efficient description of the failure mechanism at use conditions. There is a large amount of work that has been done on ALT planning. For an overview of this work, see, for example, *Nelson* ([22.1], Chapt. 6) and *Meeker* and *Escobar* ([22.3], Chapt. 20 and Sect. 22.5).
- The statistical analysis of ALT data depends importantly on the acceleration model and distribution assumptions. A sensitivity analysis should be conducted to study how results (e.g., quantile and failure probability estimates) vary with changes in the assumed model and distribution. The results of such sensitivity analyses can help decision-makers understand whether model assumptions are conservative or not.

22.4 Analysis of Accelerated Life Test Data

This section discusses the statistical analysis of ALT data. First, we outline a useful strategy to follow when choosing and fitting a model to ALT data and assessing the adequacy of the fit. Then we illustrate the proposed strategy with several different ALT applications having either one or two accelerating variables.

22.4.1 Strategy for ALT Data Analysis

This section outlines some guidelines for analyzing ALT data. Suppose that groups of data were collected under several individual conditions (levels or level combinations of the accelerating variables).

1. Make a scatter plot of the lifetimes versus the explanatory variable(s) to help in the identification of an appropriate relationship or model.
2. For each individual condition, make a probability plot (*Meeker* and *Escobar*, [22.3], Chapts. 6 and 21) of the data under candidate distributions and obtain ML estimates of the parameters. A distribution provides a good fit if the plotted points fall roughly along a straight line. Check the constant-σ assumption visually by comparing the slopes of the lines in

the probability plots and statistically by comparing the estimates of σ.

3. Fit a model constraining σ to be the same at each level and compare with the unconstrained model in Step 3. This provides the basis for a formal test of constant σ.
4. Based on the previous steps, choose and fit and overall model (distribution and acceleration model).
5. Compare the overall model fit in Step 4 to the common-σ model fit in Step 3. Large discrepancies between these two models suggest an inadequate fit for the overall model.
6. Use diagnostics such as residual plots to assess the fit of the overall model and to check the model assumptions.

22.4.2 Data Analysis with One Accelerating Variable

This section describes the analysis of several different kinds of ALT data sets, following and illustrating the ALT analysis strategy outlined in Sect. 22.4.1.

ALT Data Scatter Plot

The first step is to make a scatter plot of the data and study how the accelerating variable affects the lifetime. Use a different plotting symbol to indicate censored observations.

Example 22.5: Scatter Plot of the GAB Insulation Data. Example 22.2 described a generator armature bar insulation ALT. A scatter plot of the data, given in Fig. 22.2, suggests a downward trend in lifetime as voltage stress is increased. There was heavy censoring at the lower voltage stresses. In particular, 8 out of 12 units failed at 190 V/mm and none of the 12 units failed at 170 V/mm.

Multiple Probability Plots at Individual Conditions of the Accelerating Variable

For each level (individual condition) of the accelerating variable(s), compute a nonparametric estimate of the fraction failing as a function of time and plot it on probability paper for a suggested lifetime distribution. The distribution adequately describes the data if, for each individual condition, the points lie close to a straight line. ML can be used to fit a line through the points at each level. A multiple probability plot showing these nonparametric estimates and fitted lines for all of the levels of the accelerating variable provides a convenient visual assessment of the constant-σ assumption. If the assumption is reasonable, the lines will be approximately parallel. Repeat this for different lifetime distributions.

Example 22.6: Probability Plots at Individual Conditions for the GAB Insulation Data. Figures 22.5 and 22.6 give lognormal and Weibull multiple probability plots, respectively, for the GAB insulation data. There is nothing plotted for 170 V/mm because there were no failures at this level of voltage stress. The points for each voltage stress level fall roughly along a straight line, and the lines appear to be reasonably parallel (but more parallel with the Weibull distribution). It appears that both the Weibull and the lognormal distributions pro-

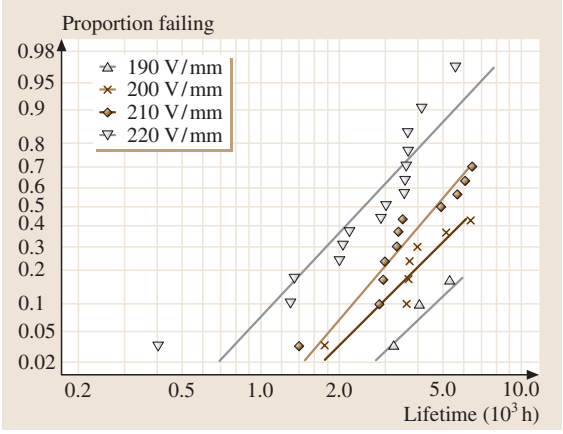


Fig. 22.5 GAB insulation data. Lognormal multiple probability plot with lognormal ML fits for each level of voltage stress

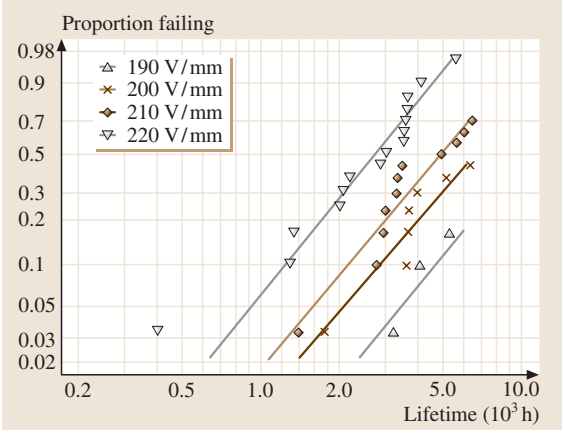


Fig. 22.6 GAB insulation data. Weibull multiple probability plot with Weibull ML fits for each level of voltage stress

Table 22.2 GAB insulation data. Weibull ML estimates for each voltage stress

	Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
190	μ	2.49	0.44	[1.63, 3.35]
	σ	0.42	0.23	[0.14, 1.26]
200	μ	2.07	0.20	[1.68, 2.45]
	σ	0.45	0.16	[0.23, 0.89]
210	μ	1.74	0.13	[1.49, 2.00]
	σ	0.44	0.11	[0.26, 0.73]
220	μ	1.17	0.11	[0.94, 1.39]
	σ	0.42	0.09	[0.28, 0.64]

The individual maximum log likelihoods were $\mathcal{L}_{190} = -12.31$, $\mathcal{L}_{200} = -22.09$, $\mathcal{L}_{210} = -27.54$, and $\mathcal{L}_{220} = -25.05$. The total log likelihood for this model is $\mathcal{L}_1 = -86.99$

vide adequate descriptions of the GAB data. At each level of voltage stress, the Weibull distribution is fitted to the data, and estimates of μ and σ were computed by using the method of ML. Table 22.2 gives the ML estimates of μ and σ for each level of voltage stress. Table 22.2 also provides approximate standard errors and normal-approximation-based confidence intervals. The lines drawn in Fig. 22.6 represent the ML estimates of the individual CDFs (fraction failing) at each level of voltage stress.

The ML estimates of σ for all levels of voltage stress are similar. The similarities are reflected in the near-parallel CDF lines drawn in Fig. 22.6. Although details are not shown here, the maximum total log likelihood value achieved by fitting Weibull distributions with common σ (the location parameters are allowed to float) to each individual condition is $\mathcal{L}_2 = -87.01$ which is very close to the total log-likelihood $\mathcal{L}_1 = -86.99$ obtained from fitting separate Weibull distributions. This suggests that it is reasonable to assume that σ is constant across voltage stresses.

Maximum Likelihood Estimate of the ALT Model

The inverse power relationship is often used to relate the lifetime distribution of a dielectric to voltage stress. Suppose that $T(\text{volt})$, the lifetime at volt, has a log-location-scale distribution with parameters (μ, σ) and that σ does not depend on volt. Under the inverse power relationship, $\mu = \beta_0 + \beta_1 x$ where $x = \log(\text{volt})$. Fitting the model to data is accomplished by computing estimates for the parameters β_0, β_1, and σ.

Example 22.7: Maximum Likelihood Estimate of the Inverse Power Model for the GAB Insulation Data. The results of fitting the inverse power relationship Weibull model to the data are summarized in Table 22.3. A likelihood ratio test comparing with the constant-σ/floating-μ model allows a formal assessment of whether the inverse power relationship is consistent with the data. Fitting the inverse power relationship with constant σ also allows extrapolation to the use level of 120 V/mm.

Suppose that model B is obtained by imposing constraints on the parameters of model A (e.g., the constraint that the location parameters at different levels of voltage stress are related by the inverse power relationship). For $i = \text{A, B}$, let $\mathcal{L}_i$ be the maximum total log likelihood achieved by fitting model i to the data. If model B is the correct model, in large samples, the likelihood ratio statistic $\mathcal{W} = 2 \times (\mathcal{L}_\text{A} - \mathcal{L}_\text{B})$ has a χ^2_ν distribution with degrees of freedom ν equal to the difference between the number of estimated model parameters in the two models. Large values of $\mathcal{W}$ indicate evidence against the constrained model model B, relative to the

Table 22.3 GAB insulation data. ML estimates for the inverse power relationship Weibull regression model

Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
β_0	53.39	9.46	[34.84, 71.93]
β_1	−9.68	1.77	[−13.14, −6.21]
σ	0.44	0.06	[0.33, 0.58]

The maximum log likelihood for this model is $\mathcal{L}_3 = -87.72$

unconstrained model A. The difference is statistically important at the α level if $\mathcal{W} \geq \chi^2_{(1-\alpha;\nu)}$ where $\chi^2_{(1-\alpha;\nu)}$ is the $(1-\alpha)$ quantile of the χ^2_ν distribution.

For the GAB insulation data, model A is the (unconstrained) set of Weibull distributions with common σ (five parameters), and model B is the (constrained) inverse power relationship Weibull model (three parameters). Then $\mathcal{L}_A = \mathcal{L}_2 = -87.01$, $\mathcal{L}_B = \mathcal{L}_3 = -87.72$, and $\mathcal{W} = 1.42 < \chi^2_{(0.95;2)} = 5.99$. Thus, the difference between the fits is not statistically important at the 0.05 level and the data do not provide evidence against model B.

The maximum log likelihood value attained with the lognormal distribution is -89.52, which is close to $\mathcal{L}_3 = -87.72$ for the Weibull distribution. Again, as noted earlier with the probability plots, both the Weibull and lognormal models fit the data well.

Figure 22.7 is a probability plot showing the ML fit of the inverse power relationship Weibull model to the data. It also gives the ML estimate of the fraction failing at use conditions $\text{volt}_U = 120\,\text{V/mm}$. The dashed lines give pointwise normal-approximation 95% confidence intervals for the fraction failing at use conditions. Conversely, this plot can be used to obtain an estimate and an approximate confidence interval for specified quantiles at use conditions. Figure 22.8 is a scatter plot of the GAB data with ML estimates of the 0.1, 0.5, and 0.9 quantiles at all levels of voltage stresses between 120 and 220 V/mm. The horizontal line indicates the censoring time at 6480 h. Densities (smallest extreme-value densities because time is plotted on a log scale) are shown at the voltage stresses in the experiment as well as at the use condition of 120 V/mm. Figure 22.8 suggests that it is unlikely that one would observe any failures below 170 V/mm during the ALT experiment.

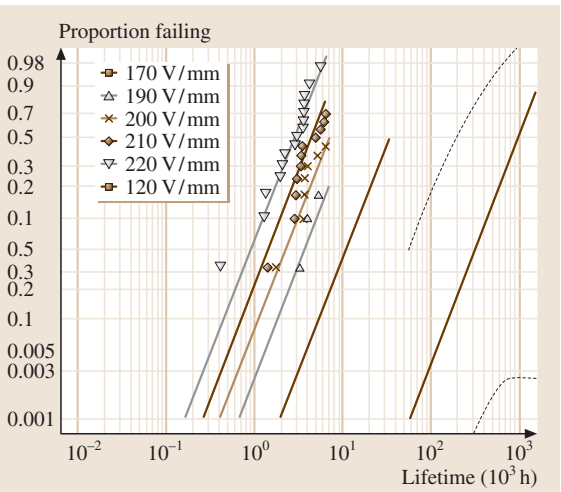


Fig. 22.7 GAB insulation data. Weibull multiple probability plot depicting the inverse power relationship Weibull regression model ML fit

Assessing Statistical Uncertainty

An estimate of the variance–covariance matrix for the ML estimates $\hat{\boldsymbol{\theta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma})'$ is

$$\widehat{\sum}_{\hat{\theta}} = \begin{pmatrix} 89.52 & -16.73 & 0.27 \\ -16.73 & 3.13 & -0.05 \\ 0.27 & -0.05 & 0.004 \end{pmatrix} . \qquad (22.11)$$

The elements of this matrix are used to compute normal-approximation confidence intervals, as described in Example 22.9. This matrix is usually available (at least as an option) as output from computer software and is useful for computing the variance of functions of the parameters (e.g., quantiles and fraction failing estimates) that may be of interest. In modern software for life data analysis, however, such computations are easy to request and printing the variance–covariance matrix is often suppressed by default.

Diagnostics and Assessing Fit of ALT Model to Data

Plots like Figs.22.7 and 22.8 can be used to assess the fit of a model to a set of data. Graphical analyses of the standardized residuals, $\exp[\log(t) - \hat{\mu}]/\hat{\sigma}$, provide additional diagnostics for judging model adequacy.

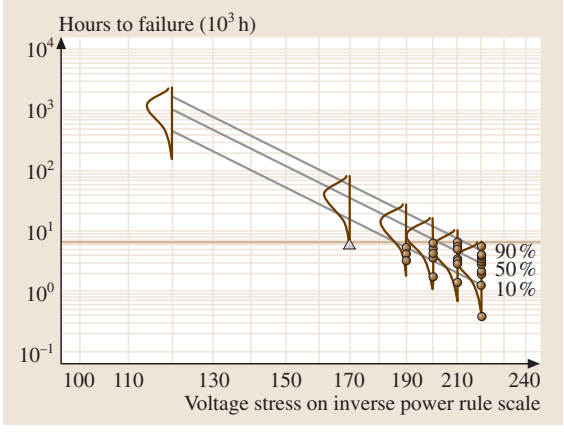


Fig. 22.8 GAB insulation data. Scatter plot showing hours to failure versus voltage stress (log–log scales) and the inverse power relationship Weibull regression model fit. Censored observations are indicated by Δ

Residuals can be plotted against the order of the test runs, fitted values, and levels of explanatory variables that are or are not in the fitted model. It is necessary to take into account the amount of censoring during interpretation. An overall assessment of the distributional assumption can be obtained by using a probability plot of the residuals.

Example 22.8: Residuals Plots for the Inverse Power Relationship Weibull Model Fit to the GAB Insulation Data. Figure 22.9 plots residuals versus fitted values. The vertical axis shows the standardized residuals. The horizontal lines are the 0.05, 0.50, and 0.95 quantiles of a standard exponential distribution (e.g., $-\log(1-0.95) = 2.996$ is the 0.95 quantile of this distribution). For the Weibull distribution, the residuals should have an approximate standard Weibull ($\mu = 0, \sigma = 1$) distribution (which is left-skewed). Thus, the Weibull distribution is appropriate if the residuals have a long lower tail. At 210 V/mm, where all of the test units failed, the residuals (first column of points in Fig. 22.9) indicate a left-skewed distribution about the center line, indicating no lack of fit of the Weibull distribution. The interpretation of the residuals is not as simple for the levels 170, 190, 200, and 210 V/mm (the last four columns of points in Fig. 22.9) because all of these levels resulted in some censored observations. The triangles (△) in Fig. 22.9 indicate lower bounds for residuals for these censored observations (i. e., the actual unobserved residuals corresponding to these points would be larger than the plotted triangles). With this in mind, there is no strong evidence of lack of fit of the Weibull distribution at these levels. Figure 22.10 is a Weibull probability plot of the residuals for all failures. The residuals fall close to a straight line, which suggests that there is no problem with the Weibull assumption.

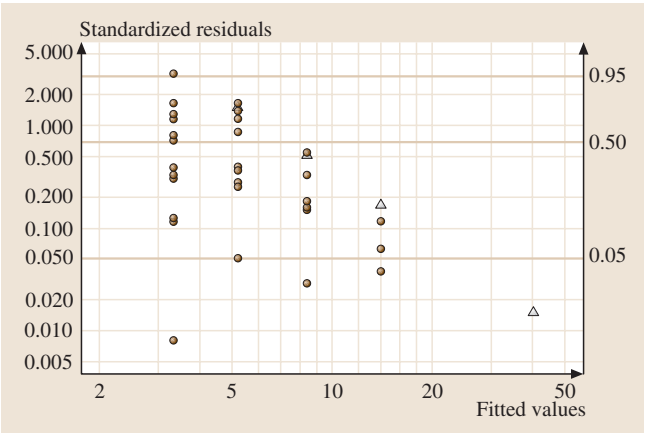


Fig. 22.9 GAB insulation data. Plot of standardized residuals versus fitted values for the inverse power relationship Weibull regression model. Censored observations are indicated by Δ

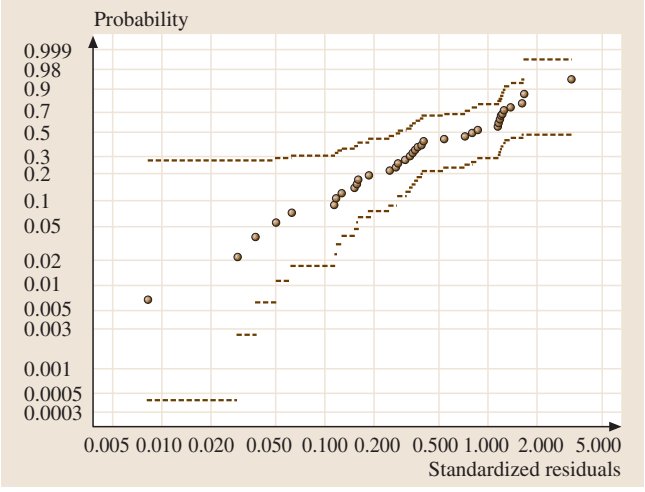


Fig. 22.10 GAB insulation data. Weibull probability plot of the standardized residuals from the inverse power relationship Weibull regression model fit

Estimation at Use Conditions and Sensitivity to Model Assumptions

It is important to quantify the statistical uncertainty in estimates from an ALT. Here, we illustrate the computation of normal-approximation confidence intervals for quantiles. See Sect. 17.4.2 of *Meeker* and *Escobar* [22.3] for a more general discussion of the methodology. The following example also contains a sensitivity analysis of the estimates with respect to the assumed distribution.

Example 22.9: ML Estimates and Confidence Intervals for Quantiles for the GAB Insulation Data Life Distribution. For the GAB experiment, it was of interest to estimate the 0.01 and 0.05 quantiles of insulation life at the 120 V/mm use condition. The ML estimate of the location parameter at 120 V/mm, computed from the ML estimates for the parameters in Table 22.3, is

$$\hat{\mu} = \hat{\beta}_0 + \hat{\beta}_1 x = 53.39 - 9.68 \log(120) = 7.07 \,.$$

The ML estimate of the log p quantile is

$$\begin{aligned} \log(\hat{t}_p) &= \hat{\mu} + \log[-\log(1-p)]\,\hat{\sigma} \\ &= 7.07 + \log[-\log(1-p)] \times 0.44 \end{aligned}$$

Table 22.4 GAB insulation data. Quantiles ML estimates at 120 V/mm

Distribution	Quantile	ML estimate	Normal-approximation 95% confidence interval
Weibull	t_{01}	154.6×10^3 h	[26.4, 907.4] $\times 10^3$ h
		17.7 y	[3.0, 103.6] y
	t_{05}	317.0×10^3 h	[51.4, 1953.0] $\times 10^3$ h
		36.2 y	[5.9, 223.0] y
Lognormal	t_{01}	254.4×10^3 h	[43.4, 1492.9] $\times 10^3$ h
		29.1 y	[5.0, 170.0] y
	t_{05}	390.8×10^3 h	[63.4, 240.9] $\times 10^3$ h
		44.6 y	[7.0, 27.5] y

with an estimated standard error $\widehat{\text{se}}_{\log(\widehat{t}_p)} = \sqrt{\widehat{\text{Var}}_{\log(\widehat{t}_p)}}$ $= \sqrt{\boldsymbol{c}' \widehat{\sum}_{\widehat{\boldsymbol{\theta}}} \boldsymbol{c}}$, where $\boldsymbol{c} = \{1, \log(\text{volt}), \log[-\log(1-p)]\}'$ and $\widehat{\sum}_{\widehat{\boldsymbol{\theta}}}$ is given in (22.11). A $100(1-\alpha)\%$ normal-approximation confidence interval for t_p is

$$\exp\left[\log(\widehat{t}_p) \pm z_{(1-\alpha/2)} \times \widehat{se}_{\log(\widehat{t}_p)}\right] = \widehat{t}_p \exp\left(\pm z_{(1-\alpha/2)} \times \widehat{se}_{\log(\widehat{t}_p)}\right) ,$$

where z_γ is the γ quantile of the standard normal distribution. Table 22.4 gives the ML estimates and 95% confidence intervals for the 0.01 and 0.05 quantiles at 120 V/mm. the data that was almost as good as the fit with the Weibull distribution. Table 22.4 also includes quantile estimates for the lognormal distribution. Although the differences between the Weibull and lognormal distribution point estimates may be judged to be large, the confidence interval for the quantile under one distribution contains the point estimate under the other. The lognormal distribution quantile estimates are optimistic relative to those for the Weibull distribution.

22.5 Further Examples

This section illustrates the use of the statistical methods, outlined in Sect. 22.4, to various other ALT applications. The examples show the fit of ALT models to interval-censored data and reliability data with two or

Table 22.5 IC device data

Temperature (°C)	Time Bounds (h) Lower	Upper	Frequency	Censoring
150	788		1	Right
150	2304		49	Right
175	788		1	Right
175	2304		49	Right
200	788		1	Right
200	2304		49	Right
250	384	788	1	Interval
250	788		1	Right
250	1536	2304	6	Interval
250	2304		42	Right
300	96	192	1	Interval
300	192	384	6	Interval
300	384	788	20	Interval
300	384		2	Right
300	788	1536	16	Interval
300	1536		5	Right

more explanatory variables. These examples illustrate a wider (but not exhaustive) range of data types that arise in practice, provide more insight into how to develop ALT models, address the goals and objectives of the experiment, and study the effect of model assumptions on these aims (sensitivity analyses).

22.5.1 Analysis of Interval Censored ALT Data

Interval-censored data are common in ALT studies. Constraints on resources limit the frequency of inspection times in tests. As a consequence, engineers often fail to observe failures instantaneously, and inspection times serve as bounds for lifetimes.

Example 22.10: Analysis of Integrated Circuit Device ALT Data. Table 22.5 gives the results of an ALT for an integrated circuit (IC) device at five different levels of junction temperature. Because the diagnostic testing of each device was expensive, each unit could only be inspected a few times during the ALT. Consequently, an observation was either interval-censored (bounds for lifetime were recorded) or right-censored (lower bound for lifetime was recorded). The engineers chose inspection times that were approximately equally spaced in log time. Tests were conducted at 150, 175, 200, 250, and 300 °C on a total of 250 devices (50 at each level of temperature).

The objectives of the ALT was to quantify the long-term reliability of the ICs and to estimate the effective activation energy of the failure mechanism. In particular, the engineers involved in the study wanted to assess the hazard rate and the fraction failing by 100 000 hours at the use junction temperature of 100 °C.

Figure 22.11 is a lognormal probability plot of the data at individual conditions. There are no plots for 150, 175, 200 °C because there were no failures at these levels of temperature. The ML results for estimating the lognormal parameters at individual temperatures are given in Table 22.6. The fitted CDF lines drawn in the plot appear to be reasonably parallel. Also, the ML estimates of σ and corresponding confidence intervals do not indicate evidence against the constant-σ assumption across temperature levels. Table 22.7 gives the ML results of fitting an Arrhenius lognormal model to the IC device data. An estimate of the effective activation energy is $\widehat{E}_a = \widehat{\beta}_1 = 0.94$.

Figure 22.12 is a probability plot depicting the ML fit of the Arrhenius lognormal model to the data. It also gives the ML estimate of the fraction failing at $\text{temp}_U = 100\,°\text{C}$ with pointwise normal-approximation 95% confidence intervals. Figure 22.12 plots the ML estimates of the 0.1, 0.5, and 0.9 quantiles with the data for temperatures between 100 and 300 °C. Even though there is a large amount of statistical uncertainty, Figs. 22.12 and 22.13 both suggest that it is very unlikely that failures would occur before 100 000 hours at $\text{temp}_U = 100\,°\text{C}$.

Observing failures at two or more levels of the accelerating variable is necessary to estimate the parameters of the Arrhenius relationship. There are situations in which the activation energy E_a is specified, based

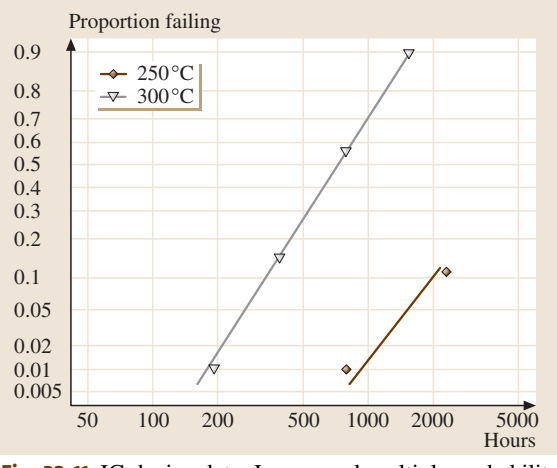


Fig. 22.11 IC device data. Lognormal multiple probability plot with individual lognormal ML fits for each temperature

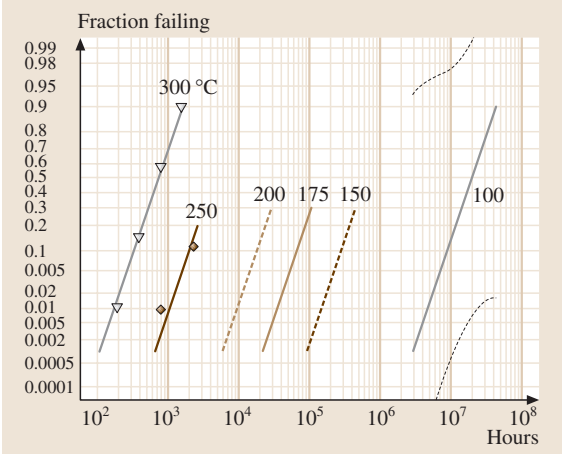


Fig. 22.12 IC device data. Lognormal multiple probability plot depicting the Arrhenius lognormal regression model ML fit

Table 22.6 IC device data. Lognormal ML estimates for each temperature

Temperature (°C)	Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
250	μ	8.54	0.34	[7.87, 9.22]
	σ	0.74	0.25	[0.38, 1.43]
300	μ	6.58	0.20	[8.59, 9.36]
	σ	0.61	0.07	[0.48, 0.76]

The individual maximum log likelihoods were $\mathcal{L}_{250} = -26.11$ and $\mathcal{L}_{300} = -63.18$. The total log likelihood for this model is $\mathcal{L}_4 = -89.29$

Table 22.7 IC device data. ML estimates for the Arrhenius lognormal regression model

Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
β_0	−12.45	1.95	[−16.26, −8.63]
β_1	0.94	0.09	[0.76, 1.12]
σ	0.62	0.07	[0.50, 0.77]

The maximum log likelihood for this model is $\mathcal{L}_5 = -89.45$

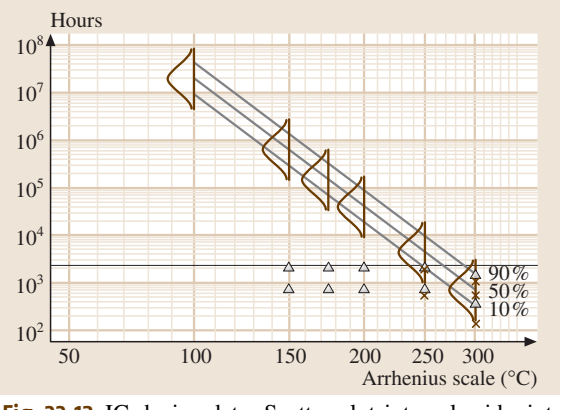


Fig. 22.13 IC device data. Scatter-plot interval midpoints failure versus °C (on an Arrhenius scale) and the Arrhenius lognormal regression model fit. Censored observations are indicated by Δ

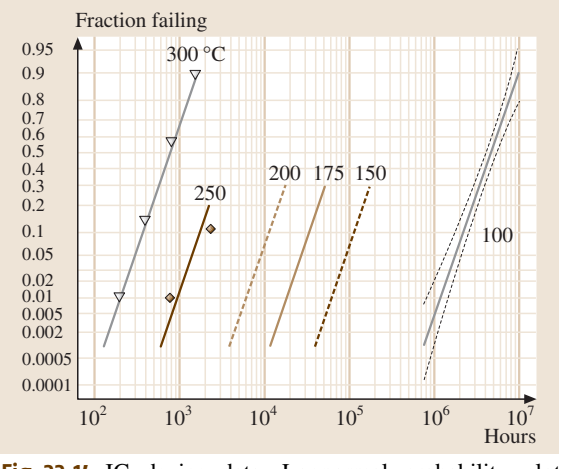


Fig. 22.14 IC device data. Lognormal probability plot showing the Arrhenius lognormal model ML estimates and 95% confidence intervals for $F(t)$ at 100 °C given $E_a = 0.8$ eV

on prior experience with the product, in which case only one temperature level with failures may suffice for estimation. For example, MIL-STD-883 [22.27] gives guidelines for conducting reliability demonstrations tests when E_a is given.

Example 22.11: Analysis of the IC Device Data when the Activation Energy is Given. Figure 22.14 is similar to Fig. 22.12, but obtained with $E_a = 0.8$ eV given. Now, the confidence intervals for the fraction failing at $\text{temp}_U = 100\,°\text{C}$ are much narrower than when E_a had to be estimated. This difference reflects a tremendous, probably unreasonable, gain in precision. Because there will be doubt about the actual value of E_a, one must exercise caution in interpreting such an analysis. Varying $\beta_1 = E_a$ over some plausible range of values provides a useful sensitivity analysis. Also a Bayesian analysis could be used when the uncertainty on E_a can be specified by a probability distribution (see *Meeker* and *Escobar* [22.3], Sect. 22.2).

22.5.2 Analysis of Data From a Laminate Panel ALT

A sample of 125 circular-holed notched specimens of a carbon eight-harness-satin/epoxy laminate panel were

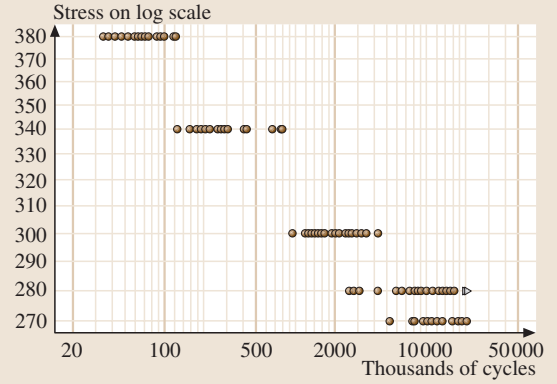


Fig. 22.15 Laminate panel data. Log–log scatter plot of life versus stress

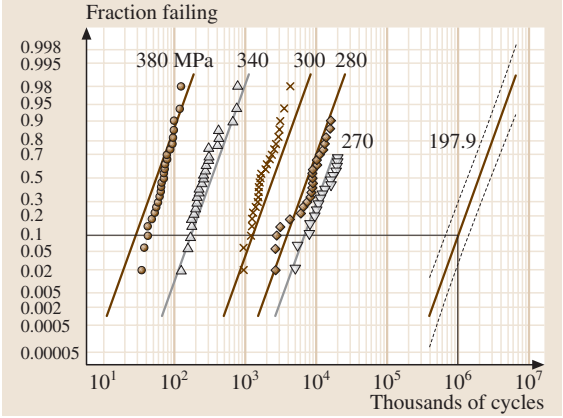


Fig. 22.16 Laminate panel data. Lognormal probability plot showing the inverse power relationship lognormal model ML fit and 95% confidence intervals for $F(t)$ at 197.9 MPa

subjected to a cyclic four-point out-of-plane bending at several levels of stress, measured in MPa. The data are from *Shimokawa* and *Hamaguchi* [22.28]. In these tests, final specimen fracture and fiber fracture occurred simultaneously. The lifetime of a specimen is the number of cycles to fracture. Figure 22.15 is a log–log scatter plot of the data. Note that the response (thousands of cycles) is plotted on the horizontal axis (the usual convention in the fatigue-fracture literature). Ten specimens were right-censored (two at 280 MPa and eight at 270 MPa). The goals of experiment were to estimate the relationship between life and stress (or the S–N curve) and a safe level of stress for a given life quantile. Figure 22.16 is a multiple lognormal probability plot of the data. The lognormal distribution appears to be appropriate here. Figure 22.15 indicates a strong linear correlation between log life and log stress. These plots suggest fitting a inverse power relationship lognormal model to the data.

Example 22.12: ML Estimate of the Inverse Power Relationship Lognormal Model for the Laminate Panel Data. For the inverse power relationship lognormal model, the parameters are $\mu = \beta_0 + \beta_1 x$ and σ, where $x = \log(\text{Stress})$. Table 22.8 gives the results of the ML estimation of β_0, β_1 and σ with the laminate panel data.

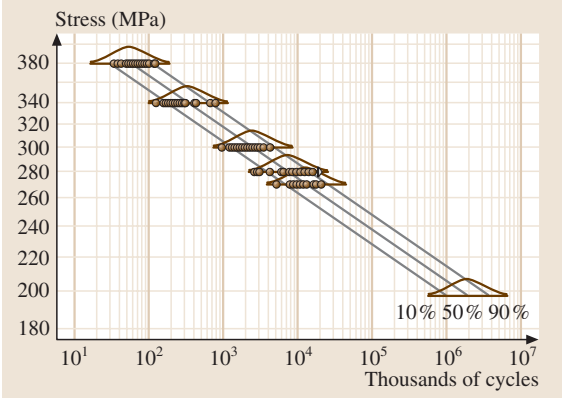


Fig. 22.17 Laminate panel data. Scatter-plot showing cycles to failure versus MPa (on a log scale) and the inverse power relationship lognormal regression model ML fit. Censored observations are indicated by Δ

Estimates of the fraction failing at the individual stress levels of the experiment are drawn in Fig. 22.16. Figure 22.17 is a scatter plot of the data with ML estimates of the 0.1, 0.5, and 0.9 quantiles. The confidence intervals for the life distribution at use conditions in this

Table 22.8 Laminate panel data. ML estimates for the inverse power relationship lognormal regression model

Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
β_0	99.36	2.14	[95.16, 103.55]
β_1	−16.05	0.37	[−16.78, −15.32]
σ	0.52	0.04	[0.46, 0.60]
The maximum log likelihood for this model is $\mathcal{L}_6 = -898.3$			

example are much narrower, when compared to those for the IC device example shown in Fig. 22.12. This is because there is less censoring and less relative extrapolation in the inference for the laminate panel data. The median (0.50 quantile) life line in Fig. 22.17 provides the estimate of the S–N curve that was of interest to the engineers.

Example 22.13: Estimate of a Safe-Level Stress for the Laminate Panel Data. The engineers wanted to estimate a safe level of stress at which no more than 10% of the population of laminate panels would fail before 1 000 000 cycles. This involves determining the stress level at which the 0.10 quantile is 1 000 000 cycles. At any stress below this safe level of stress, the population fraction failing before 1 000 000 cycles will be more than 0.10. Using the inverse power relationship lognormal model, the safe stress level is

$$\text{safe stress} = \exp\left\{\left[\log\left(10^6\right) - 99.36 - \Phi_{\text{nor}}^{-1}(0.1) \times 0.52\right] / (-16.05)\right\} = 197.9\,\text{MPa}.$$

Figure 22.16 plots an estimate of the fraction failing (rightmost line) at this safe stress level. As seen in this plot, the stress 197.9 MPa is an extrapolation because the estimation is outside the range of the data. Of course, there is potential danger in making decisions based on this estimate, especially if the inverse power relationship lognormal model fails to hold below 270 MPa. Because the inverse power relationship is an empirical model and its use is based on past experience, it is important to perform a sensitivity analysis by computing safe-level estimates under alternative relationships. For example, the reciprocal relationship ($x = 1/\text{Stress}$) gives a model maximum log likelihood value of $\mathcal{L}_7 = -892.1$ which is larger than that from the inverse power relationship (indicating that the reciprocal model provides a better fit to the data). The estimate of the safe level of stress under the reciprocal relationship is equal to 216 MPa. This is larger (less conservative) than that under the inverse power relationship. Quantile estimates at a given stress level are also less conservative under the reciprocal relationship. That is, the estimates of life under the reciprocal relationship are longer than those under the inverse power relationship.

22.5.3 Analysis of ALT Data with Two or More Explanatory Variables

The single-variable analyses in Sects. 22.4, 22.5.1, and 22.5.2 can be extended to two or more variables (accelerated or otherwise). For example, ATs with multiple accelerating variables often use temperature in combination with humidity, pressure, and/or voltage. The goal of an AT may be to compare new and old product designs under different humidity levels, using temperature to accelerate the lifetimes. In this case, product design is a qualitative factor and the humidity range would mimic that seen in the actual application. This section illustrates the analysis of data from AT studies like these.

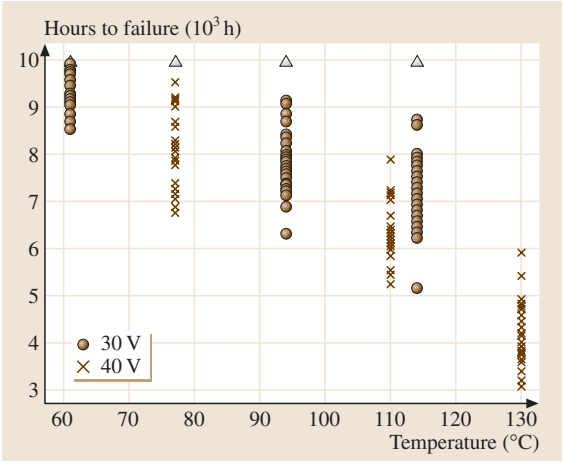


Fig. 22.18 LED device data. Scatter-plot showing hours to failure versus temperature with voltage indicated by different symbols. Censored observations are indicated by Δ

Example 22.14: Light Emitting Device Data with Accelerated Current and Temperature. Recall the LED device data from Example 22.4. The engineers were interested in the reliability of the LED at use conditions of 20 mA current and 40 °C temperature. Figure 22.18 plots the LED device data versus temperature for different levels of current. The plot suggests that failures occur earlier at higher levels of current or temperature.

The probability plot in Fig. 22.4 suggests that the lognormal distribution can be used to describe the lifetimes at each combination of current and temperature. Furthermore, because the lines are approximately parallel, it is reasonable to assume that σ is constant across individual conditions.

The inverse power (for current) and Arrhenius (for temperature) relationships are combined to model the effect that current and temperature have on lifetime. In particular, suppose that LED lifetime at volt and temp°C has a lognormal distribution with parameters μ given by

Main effects without interaction model:

$$\mu = \beta_0 + \beta_1 x_1 + \beta_2 x_2\,, \tag{22.12}$$

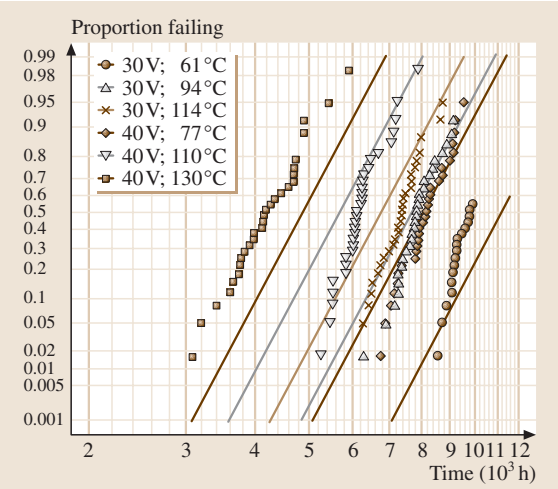


Fig. 22.19 LED device data. Multiple lognormal probability plot depicting the regression model (22.12) (without interaction) ML fit

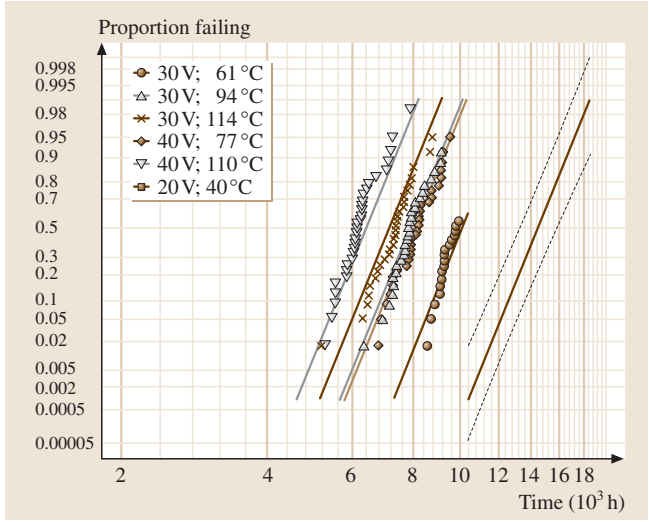


Fig. 22.20 LED device subset data. Multiple lognormal probability plot depicting the lognormal regression model (22.12) (without interaction) ML fit. The data at (40 V, 130 °C) were omitted

Main effects with interaction model:

$$\mu = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 \,, \qquad (22.13)$$

where $x_1 = \log(\text{volt})$, $x_2 = 11605/(\text{temp K})$, and $\beta_2 = E_a$, and σ does not depend on $\boldsymbol{x}$. Fitting the model with interaction to the data resulted in life estimates at use conditions that were shorter than those observed at the more stressful test conditions. The engineers were sure this was wrong. It turned out that the incorrect estimates were caused by the interaction term in the model. The interaction term lead to nonlinearity in the response surface and resulted in a saddle point in the surface relating μ to the transformed explanatory variables. Because this model gave nonsense results, it will not be considered any further.

The fit of the main-effects model without interaction reveals strong lack of fit at the high-level combination of volt = 40 and temp°C = 130. Figure 22.19 is a probability plot depicting the ML estimates of the fraction failing as a function of time at individual conditions. Observe that the true fraction failing as a function of time is underestimated throughout the range of data at volt = 40 and temp°C = 130. If the main-effects model were physically appropriate, it appears that it may not hold when both volt and temp are at these high levels, probably because a new failure mechanism is excited at those extreme conditions.

Table 22.9 provides the ML results for the inverse-power Arrhenius lognormal model after deleting the

Table 22.9 LED device subset data. ML estimates for the lognormal regression models (22.12) and (22.13)

Model	Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
Main effects without interaction	β_0	1.33	0.28	[0.78, 1.89]
	β_1	−0.46	0.06	[−0.58, −0.34]
	β_2	0.073	0.005	[0.063, 0.083]
	σ	0.11	0.07	[0.09, 0.12]
Main effects with interaction	β_0	13.54	4.20	[5.31, 21.78]
	β_1	−3.96	1.21	[−6.33, −1.60]
	β_2	−0.31	0.13	[−0.57, −0.05]
	β_3	0.11	0.04	[0.04, 0.18]
	σ	0.10	0.006	[0.09, 0.12]

The maximum log likelihood values are $\mathcal{L}_8 = -173.5$ (main effects without interaction) and $\mathcal{L}_9 = -169.5$ (main effects with interaction model)

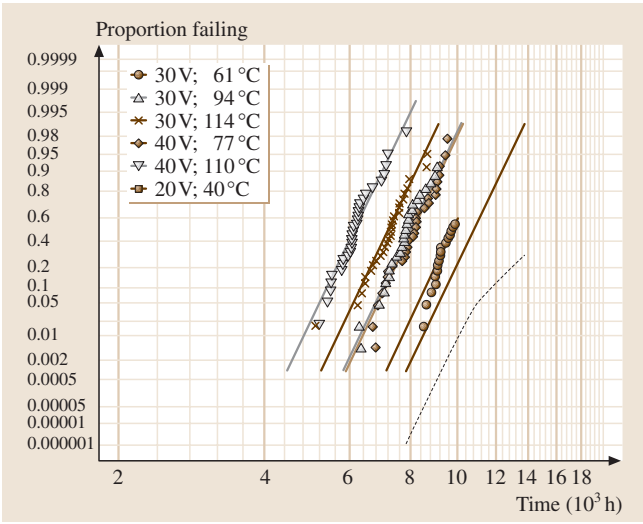


Fig. 22.21 LED device subset data. Multiple lognormal probability plot lognormal regression model (22.12) (with interaction) ML fit. The data at (40 V, 130 °C) were omitted

data at volt = 40 V and temp°C = 130 °C. Figure 22.20 shows a probability plot of the LED device subset data along with ML estimates of fraction failing as a function of time for the model without interaction.

Figure 22.20 is a lognormal probability plot of the LED device subset data along with ML estimates of fraction failing as a function of time. It also includes the ML estimate of the fraction failing as a function of time at the use conditions volt = 20 V and temp°C = 40 °C (the rightmost line). The dashed lines represent approximate 95% pointwise confidence intervals for the fraction failing as a function of time at these conditions. For this model there is much better agreement between the model and the data. Figure 22.21 is a similar multiple probability plot using the interaction model. Comparison of the models with and without the interaction term, after deleting the (40 V, 130 °C) condition, indicates some lack of fit for the model without interaction (compare the log likelihoods in Table 22.9). Predictions with the interaction model, however, are very pessimistic and much worse than had been expected by the engineers (though not as easily dismissed as the with the interaction model when fitted to the complete data set). The estimates at use conditions are highly sensitive to the model choice because of the rather large amount of extrapolation involved in computing the estimates. The engineers were uncertain as to the correct model, but felt that almost certainly the truth would lie between the two different fits shown in Figs. 22.20 and 22.21.

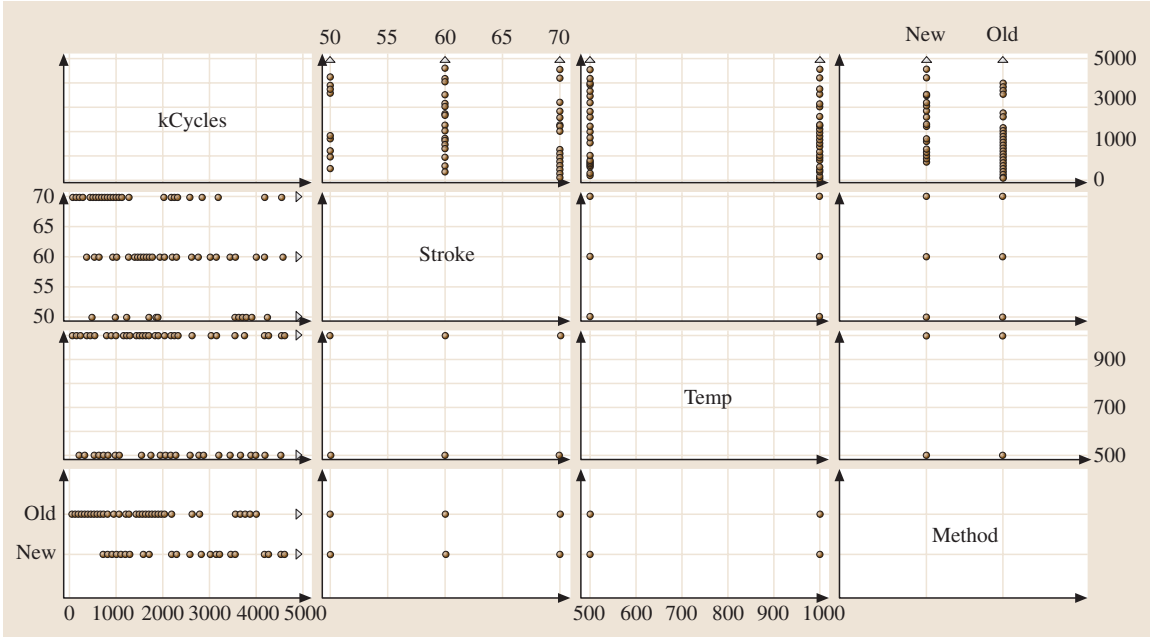


Fig. 22.22 Spring fatigue data. Scatter-plot matrix of the data

Example 22.15: An Accelerated Life Test for a Spring with Three Explanatory Variables. *Meeker*, *Escobar*, and *Zayac* [22.29] describe an experiment to compare old and new processing methods for manufacturing a spring. The experimenters used three variables, namely, stroke displacement (Stroke), processing method (Method), and temperature (temp). Springs were compressed cyclically and Stroke measured the distance that a spring was compressed. Stroke was the accelerating variable. Lifetime was measured in cycles to failure. The springs were designed to be used at Stroke= 20 mils (1 mil =1/1000 inch) and the nominal processing temperature was temp = 600 °F. The goals of the experiment were to compare the old and new processing methods, to see if temperature had an important effect on spring life, and to check if the B10 life (0.10 quantile) at use conditions was at least 500 000 kilocycles.

Meeker, *Escobar*, and *Zayac* [22.29] give the results from a 2×2×3 factorial with factors Stroke (50 mil, 60 mil, 70 mil), temp (500 °F, 1000 °F), and Method (Old, New). Each of the 12 factor combinations was replicated nine times for a total of 108 observations. All nine observations at Stroke= 60, temp = 500 °F, and Method= New were right-censored at 500 000 cycles. Figure 22.22 is a scatter-plot matrix of the Spring ALT data. It appears that on average, spring lifetimes are longer at shorter stroke, at lower temperature, and with the new processing method, respectively.

Figure 22.23 gives Weibull probability plots of the data at individual conditions (factor combinations). The plot indicates that the Weibull distribution can be used to describe the data. Moreover, the plotted lines showing the ML estimates of fraction failing as a function of time at each test condition suggest that it is reasonable to assume that σ (or, equivalently, the Weibull shape parameter β) is constant. The likelihood ratio test for homogeneity of the Weibull shape parameters provides no evidence of differences. Thus, one can assume constant σ across experimental factor combinations.

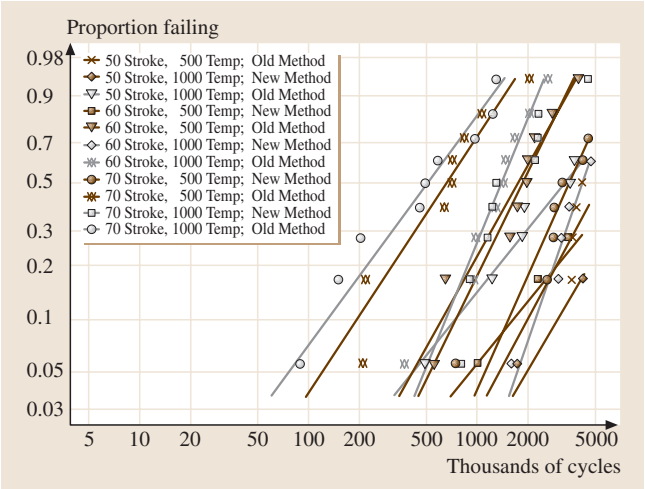


Fig. 22.23 Spring fatigue data. Weibull multiple probability plot with individual Weibull ML fits for each test condition

The model for the location parameter μ was developed as follows. Previous experience with testing similar springs suggested taking a log transformation of Stroke. On the other hand, no transformation is applied to temp because there were no prior studies involving temp (note that because temp is a processing temperature rather than a temperature driving a chemical reaction, there is no suggestion from physical/chemical theory that the Arrhenius relationship should be used). An indicator variable was used to represent the effect of the qualitative factor Method. Thus,

$$\mu = \beta_0 + \beta_1 \log(\text{Stroke}) + \beta_2\, \text{temp} + \beta_3\, \text{Method} \tag{22.14}$$

where Method = 0 for the New processing method and Method = 1 for the Old method. Terms representing factor interactions can be added to (22.14). *Meeker* et al. [22.29] extended (22.14) to include two-factor interaction terms. The likelihood-ratio test suggested,

Table 22.10 Spring fatigue data. ML estimates for the Weibull regression model

Parameter	ML estimate	Standard error	Normal-approximation 95% confidence interval
β_0	32.03	2.48	[27.16, 36.90]
β_1	−5.51	0.59	[−6.66, −4.36]
β_2	−0.000 88	0.000 27	[−0.0014, −0.000 35]
β_3	−1.27	0.15	[−1.56, −0.98]
σ	0.57	0.054	[0.47, 0.69]

The maximum total log likelihood for this model is $\mathcal{L}_{10} = -625.8$

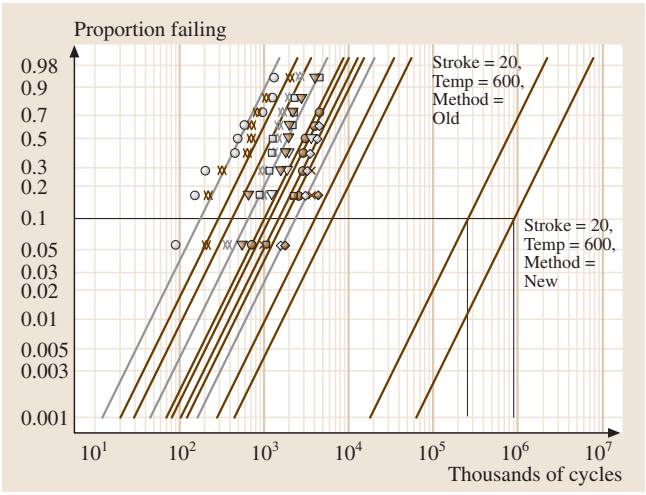


Fig. 22.24 Spring fatigue data. Weibull multiple probability plot of the Weibull regression ML fit with estimates at use conditions

however, that there was little evidence for such interactions.

Table 22.10 gives the results of fitting the Weibull model with the location parameter given in (22.14) and constant σ. For this fit, Fig. 22.24 is a probability plot showing the ML estimates for fraction failing as a function of time for each of the 12 factor combinations and for the Old and New processing methods at use conditions of Stroke = 20 mils and temp = 600 °F. We note the following:

- The estimate and confidence interval for β_1, the coefficient for Stroke, are negative, which suggests that, with temp and Method held constant, increasing Stroke shortens spring life, on average. A similar analysis shows that increasing temperature decreases the life of the units.
- The estimate for the coefficient β_3 for Method is $\hat{\beta}_3 = -1.27 < 0$ and the corresponding confidence interval excludes 0. Thus, there is evidence that switching from the Old to the New method lengthens spring life, on average. In Fig. 22.24, observe that the fraction-failing estimate for the New method is to the right (suggesting longer lives) of the fraction failing for the New method. On the average, springs under the New method last $1/\exp(\hat{\beta}_3) = 3.56$ times longer than those under the Old method. A 95% confidence interval for this factor is [2.66, 4.76] which is obtained by taking the reciprocal of the antilog of the endpoints of the confidence interval for β_3 in Table 22.10.

For a graphical comparison of quantile estimates under the two processing methods, draw horizontal lines in Fig. 22.24 that intersect the New and Old fraction-failing lines, and read off the estimates from the horizontal time axis. Estimates of the 0.10, 0.50 and 0.90 quantiles for the two processing methods under use conditions are given in Table 22.11. This table also provides estimates of the standard errors of quantile estimators and 95% normal-approximation confidence intervals. Observe that quantile estimates for the New method are 3.57 times larger than those for the Old method.

The design engineers wanted to know if the B10 life at use conditions would exceed 500 000 kilocycles. Line segments are drawn in Fig. 22.24 to indicate the B10 estimates. The Old method fails to satisfy this requirement because its B10 life estimate is 252 208 kilocycles. The estimate for the New method is 900 222 kilocycles, which does satisfy the requirement. The lower endpoint of the confidence interval is, however, below the 500 000 target. Thus, there is some doubt that the New method meets the reliability requirement.

Example 22.16: Sensitivity Analysis for the Spring Fatigue Data. Meeker et al. [22.29] studied the effect that varying model assumptions has on the analysis of the spring fatigue data. In particular, they investigated the sensitivity of the estimate of B10 life to alternative transformations of Stroke and temp, and the assumed distribution. A summary of their findings follows.

Table 22.11 Spring fatigue data. Quantiles ML estimates at (20 mil, 600 °F) for the Old and New processing methods

Processing method	Quantile	ML estimate	Standard error	Normal-approximation 95% confidence interval
Old	0.10	252 208	160 148	[72 654, 875 502]
	0.50	737 371	483 794	[203 801, 2 667 873]
	0.90	1 460 875	989 321	[387 409, 5 508 788]
New	0.10	900 221	612 067	[237 471, 3 412 619]
	0.50	2 631 947	1 855 063	[661 194, 10 476 722]
	0.90	5 214 400	3 794 441	[1 252 557, 21 707 557]

Meeker et al. [22.29] extended (22.14) to

$$\mu = \beta_0 + \beta_1 \mathrm{W} + \beta_2 \mathrm{temp} + \beta_3 \mathrm{Method}\,, \qquad (22.15)$$

where

$$\mathrm{W} = \begin{cases} \dfrac{(\mathrm{Stroke})^{\lambda} - 1}{\lambda}, & \lambda \neq 0 \\ \log(\mathrm{Stroke}), & \lambda = 0 \end{cases} \qquad (22.16)$$

and where λ was allowed to range between -1 and 2. The transformation W is known as a Box–Cox transformation, and the linear (no transformation, $\lambda = 1$) and log ($\lambda = 0$) relationships are special cases of this transformation. *Meeker* et al. [22.29] used a range of fixed values of λ and fitted by ML methods a Weibull distribution with μ in (22.15) and constant σ. They observed that, as λ moved from -1 to 2, the ML estimate of the B10 life at use conditions decreased by a factor of between 10^3 and 10^4. In particular, switching from $\lambda = 0$ to $\lambda = 1$ decreased the B10 life estimate from 900 221 kilocycles to 84 925 kilocycles. From past experience with the spring, the engineers thought that λ should be close to 0 and almost certainly less than 1. Thus, a conservative approach for designing the product using the spring as a component would be to assume that $\lambda = 1$.

Using a similar sensitivity analysis with temperature, *Meeker* et al. [22.29] observed that the estimate of B10 life was not as sensitive to temp as it was to Stroke. Increasing λ from -1 to 2 increased the B10 life estimate by no more than a factor of 1.25. They also explain that the data did not provide any information about what transformation (λ value) would be best for model fitting (because there were only two levels of temperature in the experiment).

Another sensitivity analysis involved fitting a lognormal (instead of Weibull) distribution with a location parameter given by (22.15) and constant σ. For any fixed value of λ between -1 and 2, the lognormal distribution gave more optimistic estimates of B10 life than the Weibull distribution. In particular, the lognormal estimates of B10 life were about twice as large as the Weibull estimates. This is expected because, when both distributions are fitted to the same data, the Weibull distribution is less optimistic when extrapolating in time.

22.6 Practical Considerations for Interpreting the Analysis of ALT Data

Extrapolation is an integral part of any ALT. As a result, interpretation of statistical analysis of ALT data is often a challenge. In this section, we discuss the most important issues and suggest strategies to address them.

ALTs are most effective when predicting the life for well-understood failure mechanisms for relatively simple materials or components. ALTs are less useful or there may be more restrictions for more complicated products. There are also difficulties when there are discrepancies between laboratory and field conditions. In particular, when the ALT model generates failure modes that are different from those seen in the field, the information obtained from an ALT may be difficult or impossible to interpret.

ALTs have a better chance to produce useful results if they are carefully planned with attention to obtaining information about possible failure modes from which the product or material might fail.

- Information from earlier experiences with product prototypes or similar products is important. For example, pilot studies can provide information on which environmental factors and failure modes are relevant relative to customer or field use.
- Knowledge of the physical or chemical nature of product degradation and failure is crucial in providing a basis for the statistical model to be fitted to ALT data. This could provide information on how much extrapolation is justified.
- An obvious way to limit the effect of extrapolation is to test at lower levels of the accelerating variables. Doing this, however, results in fewer observed failures, which translates into statistical estimates with larger amounts of uncertainty. For one-variable ALTs, *Meeker* and *Hahn* [22.10] provided practical guidelines for choosing ALT plans that meet practical constraints and have good statistical efficiency.

22.7 Other Kinds of ATs

Discussions up to this point pertain to ALTs where products are tested at high levels of the accelerating variables and extrapolating to assess life at use levels of these variables. Of primary interest is the life distribution with

respect to a few simple failure mechanisms. This section briefly discusses other types of ATs that are used in assessing or improving product reliability. See *Meeker* and *Escobar* ([22.3], Chapt. 19) and references therein for more detailed discussion of these tests and other references.

22.7.1 Continuous Product Operation Accelerated Tests

In this type of AT, tests are performed on the complete system. Such tests are useful because, in addition to testing individual components, the tests can catch problems with component interfaces that cannot be tested in low-level ALTs. In these tests, a sample of product is tested on a more or less continuous basis to accelerate failure mechanisms. Applications include accelerated testing of engines (e.g., in automobiles, motorcycles, lawn equipment, washing machines) and refrigerator compressors. Caution is taken to avoid damage to the system that would not occur in actual product use (e.g., overheating or melted parts due to extreme temperatures). In automobile engine testing, there is one test standard in which the engine is run continuously and another in which the engine is cycled on and off continuously. This is because the two different ATs will tend to generate different kinds of failures.

22.7.2 Highly Accelerated Life Tests

In highly accelerated life tests HALT, products are operated under extremely high levels of variables such as temperature (−90 °C to 190 °C), vibration (up to 60 Grms), and other stresses directly related to system operation. These tests are used to identify failure modes and determine product operational limits. It is generally agreed that these tests provide little or no information about product life at use conditions. This is because the results of the tests usually lead to product design changes. HALTs are typically performed on solid-state electronics, but, with proper equipment, they can be applied to other products as well. *Confer* et al. [22.30] remarked that HALTs can be used to inspect lots of incoming components and to screen products by burn-in. At extremely high levels of the stressing variables, the failure-mechanism model may no longer hold or the system experiences some damage. Thus, precautions must be taken to avoid these effects. See, for example, *McLean* [22.31] and *Hobbs* [22.32] for more information on HALT methods.

22.7.3 Environmental Stress Tests

These tests are closely related to HALTs in that the main objective is to identify reliability issues quickly (e.g., product design flaws) and address them while in the product-development stage. Different authors have different names for this type of test. One is stress–life (STRIFE) tests in which one or two prototypes are aggressively tested to produce failures quickly. Most common STRIFE tests involve temperature and vibration that are cycled in increasing amplitudes throughout the test. *Nelson* ([22.1], pages 37–39) refers to such tests as 'elephant tests.' He described tests of ceramic cookware exposed to cycles of high and low temperature until failure. Failures occurring in these tests are often due to design flaws in the product. Thus, this type of test allows engineers to identify design flaws and fix them while in the design stage. Also, the engineers must determine if failures due to these flaws would occur in actual to avoid unnecessary product redesign. *Nelson* ([22.1], pages 38–39) describes an example where a transformer was redesigned to eliminate a failure mode that was discovered in an elephant test. Years later it was realized that none of the transformers with the old design ever failed from this failure mode; the expensive redesign had been unnecessary.

Nelson ([22.1], Chapt. 1) and *Meeker* and *Escobar* ([22.3], pages 519–520) give more detailed discussions of issues related to environmental stress tests.

22.7.4 Burn-In

Early failures (infant mortality) are the most common problem with electronic products. Defective components, sometimes called freaks, comprise a small fraction of all products and are caused by manufacturing defects. Burn-in involves operating the units at higher that usual levels of accelerating variables for a specific length of time to screen out such defects. For example, burn-in of products such as integrated circuits involve testing units at high levels of temperature and humidity. Units that survive this burn-in period are then put into service. This type of test is expensive because it is essentially a 100% inspection of products. Also, the longer the burn-in period, the more defective items are identified. Tradeoffs between these and other considerations are studied to determine the length of the burn-in period. *Jensen* and *Petersen* [22.33] describe some of the engineering issues related to burn-in. *Kuo* et al. [22.34] discuss statistical and economic issues of burn-in.

22.7.5 Environmental Stress Screening

Environmental stress screening (ESS) methods were developed to improve traditional burn-in tests. It has been suggested that ESS is less expensive and more effective in detecting defectives from the product population. In ESS, products are tested at milder, but more complex, stress processes than in burn-in tests. The methodology has been developed mostly along engineering considerations. The goal is still to screen out defectives without causing damage to the systems. It is widely used in consumer, industrial, and military electronics manufacturing. *Nelson* ([22.1], pages 39–40) provided references regarding ESS military standards and more in-depth discussions of ESS.

22.8 Some Pitfalls of Accelerated Testing

This section reviews potential difficulties one may face in the analysis and interpretation of ALT data. See *Meeker* and *Escobar* [22.35] for a more detailed discussion of these pitfalls.

22.8.1 Failure Behavior Changes at High Levels of Accelerating Variables

It is possible that failure mechanisms at high levels of the accelerating variables may not be the same as those at low levels. High levels of the accelerating variables could induce failure modes that have not been identified beforehand or which may not occur in normal use of the product. Such failure modes can be direct results of physical changes brought on by extreme stressing (e.g., at high temperatures product parts could warp or melt). Also, even with the same failure mechanism, the assumed relationship between life and the accelerating variable(s) may not be valid at higher levels. For example, in a voltage-accelerated ALT, the relationship between log(Life) and log(volt) may not be linear at higher levels of volt.

Previously unrecognized failure modes could still be accounted for in the data analysis by regarding failures caused by them as right-censored with respect to the failure mode of primary interest. This approach, however, requires the assumption of independent failure modes. *Nelson* ([22.1], Chapt. 7) discusses methods for analyzing failure data with competing failure modes, but the failure modes at the higher levels of the accelerating variables restrict the amount of information that will be available on the mode of interest. If possible, the investigator should establish limits on the accelerating variables within which only the primary failure mode occurs and the assumed ALT model holds.

In some applications there will be an effort to eliminate a particular failure mode that is seen only at higher levels of stress. This would be done to make it possible to study another failure mode of primary interest more effectively.

22.8.2 Assessing Estimation Variability

Estimates from ALTs will contain statistical uncertainty due to having only a limited amount of data. It is unsafe to base product-design decisions on point estimates alone. Instead, conclusions from data analysis should account for statistical uncertainty by using confidence intervals. Such confidence intervals provide a range of plausible values for quantities of interest. All of our examples have presented such confidence intervals. Recall the analysis of the GAB insulation data in Example 22.2. Estimates and corresponding pointwise 95% confidence intervals for the fraction failing at $\text{volt}_U = 120$ are shown in Fig. 22.7. The confidence interval for the probability of failure before 5×10^5 h is [0.0017, 0.94]. This interval is too wide to be of any practical value. There are ways to reduce statistical uncertainty (reduce confidence interval width). In general, one could, for instance, increase the sample size, observe more failures by testing at higher stress levels, or reduce the amount of extrapolation. One could also use prior information to fix the values of model parameters that would normally be estimated from data, if there is good information about such parameters. Recall the analyses of the IC device data in Examples 22.10 and 22.11. In the first analysis, all model parameters, including the activation energy E_a, were estimated and pointwise confidence intervals for the fraction failing at use condition temp $= 100\,^\circ\text{C}$ are given in Fig. 22.12. Figure 22.14 is a similar plot for the second analysis with E_a fixed at 0.8 eV, resulting in much narrower confidence intervals. Of course, conclusions from an analysis like that shown in Fig. 22.14 could be misleading if E_a were to be misspecified. *Meeker* and *Escobar* ([22.3], Sect. 22.2) discuss applications of Bayesian analysis

where prior distributions are set up to reflect information about model parameters.

It is important to recognize that the uncertainty reflected in confidence intervals does not include the error (bias) that results from fitting an incorrect model (and strictly speaking, no statistical model is exactly correct). More extrapolation will exacerbate the problem. So, it is important to do sensitivity analysis to study the effect that changes in the model assumptions will have on the results. For example, one could study how estimates and confidence intervals change when the form of the stress-life relationship and/or distribution is changed, as was done in several of the examples in this chapter.

22.8.3 Degradation and Failure Measured in Different Time Scales

In certain applications, lifetime is measured in different time scales (e.g., cycles or hours to failure) depending on which factors affect the degradation process. Often, time scales are determined by how the product is actually used. Incandescent light bulb burn time drives a filament-evaporation process, leading to eventual failure. One way to speed up time is by increasing voltage. The bulb is operated continuously and lifetime is measured in hours of operation. Also, turning a light bulb on and off produces mechanical shocks that can cause the initiation and growth of fatigue cracks within the filament. Bulb life in this case might be measured in on–off cycles. Testing bulbs continuously at higher voltages does not simulate the on–off cycling effect. Thus, predictions based on a simple model with only one time scale will not accurately describe light bulbs that fail through a combination of burn time and on–off cycles.

22.8.4 Masked Failure Modes

In some situations, higher levels of the accelerating variable can produce a new failure mode, not ordinarily seen at use conditions. This new failure mode can mask the failure mode of interest. If the new failure mode is not recognized (because engineers naively think they are the same and do not check), the resulting conclusions will be highly optimistic relative to the truth. See *Meeker* and *Escobar* ([22.3], Fig. 19.22) for an illustration and further discussion.

22.8.5 Differences Between Product and Environmental Conditions in Laboratory and Field Conditions

For accurate prediction of failure behavior in the field, it is necessary that laboratory conditions be similar to those in the field, and that units tested in the laboratory be similar to products in actual use. Laboratory conditions are not always similar to field conditions. For example, in a life test of a circuit pack, accelerated levels of temperature resulted in lower humidity levels. This in turn inhibited corrosion, which was the primary cause of failure in the field. Thus, there were more failures in the field than were predicted by the data analysis. Laboratory tests should have controlled both temperature and humidity levels. Also, laboratory tests are generally carefully controlled, while in actual use, the environment may be highly variable.

There are various points in the process from prototype development to product manufacture at which product-building inconsistencies could occur. Technicians may build prototypes to be used in testing in a manner that is different from the way an assembly line builds the final product. For example, products with epoxy have to be cured for a certain length of time in an oven. Uncured epoxy can be highly reactive and may cause corrosion and lead to premature failure. It is possible that the units are effectively cured in the lab, but not in the factory assembly line. ALT tests will likely predict failure behavior that is more optimistic than that in the field. Raw materials and instruments could also vary. It is also possible, because of identified design flaws, to modify the product design several times throughout the process. Any inconsistencies in how a product is built will most likely cause disagreements between laboratory and field results.

22.9 Computer Software for Analyzing ALT Data

Nelson ([22.1], pages 237–240) provides a table listing software packages for analysis of ALT data by maximum likelihood methods (which has been updated in the 2004 paperback edition). The table provides the packages' features and specific capabilities. Commercially available statistical software such as SAS JMP [22.36], SAS [22.37], MINITAB [22.38], and S-PLUS [22.39] have procedures specifically for ALT data analysis.

ALTA [22.40] is a software package specifically for ALT analysis.

All of the analyses in this chapter were done with S-PLUS Life Data Analysis (SPLIDA) by *Meeker* and *Escobar* [22.41]. SPLIDA runs on top of S-PLUS. Special features for some plots were created by modifying SPLIDA functions and using S-PLUS annotation tools. SPLIDA consists of special S-PLUS functions and GUIs, and is available for download from http://www.public.iastate.edu/~splida/. The SPLIDA download also contains files with all of the data sets from this chapter, as well as many other example data sets, including all examples from *Meeker* and *Escobar* [22.3] and many from *Nelson* [22.1].

References

22.1 W. Nelson: *Accelerated Testing: Statistical Models, Test Plans, Data Analyses* (Wiley, New York 1990)

22.2 P. A. Tobias, D. C. Trindade: *Applied Reliability*, 2nd edn. (Nostrand Reinhold, New York 1995)

22.3 W. Q. Meeker, L. A. Escobar: *Statistical Methods for Reliability Data* (Wiley, New York 1998)

22.4 W. Q. Meeker, M. Hamada: Statistical tools for the rapid development & evaluation of high-reliability products, IEEE Trans. Reliab. **44**, 187–198 (1995)

22.5 W. Q. Meeker, L. A. Escobar: Reliability: the other dimension of quality, Qual. Technol. Qual. Man. **1**, 1–25 (2004)

22.6 L. A. Escobar, W. Q. Meeker, D. L. Kugler, L. L. Kramer: Accelerated destructive degradation tests: data, models, and analysis. In: *Mathematical, Statistical Methods in Reliability*, ed. by B. H. Lindqvist, K. A. Doksum (World Scientific, Singapore 2003)

22.7 D. R. Johnston, J. T. LaForte, P. E. Podhorez, H. N. Galpern: Frequency acceleration of voltage endurance, IEEE Trans. Electr. Insul. **14**, 121–126 (1979)

22.8 N. E .Dowling: *Mechanical Behavior of Materials* (Prentice Hall, Englewood Cliffs 1993)

22.9 D. J. Klinger: Failure time and rate constant degradation: an argument for the inverse relationship, Microelectron. Reliab. **32**, 987–994 (1992)

22.10 W. Q. Meeker, G. J. Hahn: *How to Plan an Accelerated Life Test: Some Practical Guidelines*, ASQC Basic References in Quality Control: Statistical Techniques, Vol. 10 (Am. Soc. Qual. Control, Milwaukee

22.11 M. J. LuValle, T. L. Welsher, K. Svoboda: Acceleration transforms and statistical kinetic models, J. Stat. Phys. **52**, 311–320 (1988)

22.12 N. Doganaksoy, G. J. Hahn, W. Q. Meeker: Accelerated testing for speedier reliability analysis, Qual. Prog. **36**, 58–63 (2003)

22.13 K. C. Boyko, D. L. Gerlach: Time dependent dielectric breakdown of of 210 oxides, Proc. IEEE Int. Reliab. Phys. Symp. **27**, 1–8 (1989)

22.14 J. Klinger D. On the notion of activation energy in reliability: Arrhenius, Eyring, and thermodynamics, Proceedings of the Annual Reliability, Maintainability Symposium (IEEE, New York 1991) 295–300

22.15 F. M. d'Heurle, P. S. Ho: Electromigration in thin films. In: *Thin Films–Interdiffusion and Reactions*, ed. by J. M. Poate, K. N. Tu, J. W. Mayer (Wiley, New York 1978) pp. 243–303

22.16 P. B. Ghate: Electromigration–induced failures in VLSI interconnects, Proc. Int. Reliab. Phys. Symp. **20**, 292–299 (1982)

22.17 J. R. Black: Electromigration–a brief survey and some recent results, IEEE Trans. Electron. Dev. **16**, 338–347 (1969)

22.18 K. T. Gillen, K. E. Mead: *Predicting life expectancy and simulating age of complex equipment using accelerated aging techniques, National Technical Information Service* (U.S. Department of Commerce, Springfield 1980)

22.19 D. J. Klinger: Humidity acceleration factor for plastic packaged electronic devices, Qual. Reliab. Eng. Int. **7**, 365–370 (1991)

22.20 M. J. LuValle, T. L. Welsher, J. P. Mitchell: A new approach to the extrapolation of accelerated life test data, The Proc. of the Fifth Int. Conf. Reliability and Maintainability (Tech. Dig., Biarritz 1986) 620–635

22.21 G. Boccaletti, F. D'Esponosa, F. R. Borri, E. Ghio: Accelerated tests. In: *Microelectronic Reliability, Volume HI, Reliability, Integrity Assessment and Assurance*, ed. by E. Pollino (Artech House, Norwood 1989) Chapt. 11

22.22 W. B. Joyce, K-Y Liou, F. R. Nash, P. R. Bossard, R. L. Hartman: Methodology of accelerated aging, AT&T Tech. J. **64**, 717–764 (1995)

22.23 D. S. Peck: Comprehensive model for humidity testing correlation, Proc. Int. Reliab. Phys. Symp. **24**, 44–50 (1986)

22.24 D. S. Peck, C. H. Or. Zierdt: The reliability of semiconductor devices in the Bell system, Proc. IEEE **62**, 185–211 (1974)

22.25 H. Eyring, S. Gladstones, K. J. Laidler: *The Theory of Rate Processes* (McGraw Hill, New York 1941)

22.26 H. Eyring: *Basic Chemical Kinetics* (Wiley, New York 1980)

22.27 MIL-STD-883: *Test Methods and Procedures for Microelectronics* (Naval Publ. Forms Center, Philadelphia 1985)

22.28 T. Shimokawa, Y. Hamaguchi: Statistical evaluation of fatigue life and fatigue strength in

circular-holed notched specimens of a carbon eight-harness-satin/epoxy laminate. In: *Statistical Research on Fatigue and Fracture*, Cur. Mater. Res., Vol. 2, ed. by T. Tanaka, S. Nishijima, M. Ichikawa (Elsevier Applied Science, London 1987) pp. 159–176

22.29 W. Q. Meeker, L. A. Escobar, S. A. Zayac: Use of sensitivity analysis to assess the effect of model uncertainty in analyzing accelerated life test data. In: *Case Studies in Reliability and Maintenance*, ed. by W. R. Blischke, D. N. P. Murthy (Wiley, New York 2003) Chapt. 6

22.30 R. Confer, J. Canner, T. Trostle: Use of highly accelerated life test (HALT) to determine reliability of multilayer ceramic capacitors, Proc. Electron. Comp. Technol. Conf. **41**, 320–322 (1991)

22.31 H. McLean: *HALT, HASS and HASA Explained: Accelerated Reliability Techniques* (ASQ Quality, Milwaukee 2000)

22.32 G. K. Hobbs: *Accelerated Reliability Engineering: HALT and HASS* (Wiley, New York 2000)

22.33 F. Jensen, N. E. Petersen: *Burn-In: An Engineering Approach to Design, Analysis of Burn-In Procedures* (Wiley, New York 1982)

22.34 W. Kuo, W. T. K. Chien, T. Kim: *Reliability and Stress Burn-In* (Kluwer Academic, Dordrecht 1998)

22.35 W. Q. Meeker, L. A. Escobar: Pitfalls of accelerated testing, IEEE Trans. Reliab. **47**, 114–118 (1998)

22.36 SAS Institute Inc.: JMP User's Guide, Version 5.1.1 (SAS Institute, Cary 2004) http://www.jmp.com/

22.37 SAS Institute Inc.: *SAS/QC® Software: Changes and Enhancements, Release 8.2* (SAS Institute, Cary 2001) http://www.sas.com/statistics/

22.38 MINITAB: *MINITAB User's Guide 2: Data Analysis, Quality Tools* (Minitab, Inc. State College 2000) http://www.minitab.com/

22.39 SPLUS 6.2: *Guide to Statistics*, Vol. 2 (Insightful Corp., Seattle 2001) http://www.insightful.com/

22.40 ALTA: *ALTA Version 6 User's Guide* (Reliasoft, Tucson 2002)

22.41 W. Q. Meeker, L. A. Escobar, SPLIDA User's Manual (Iowa State Univ., Ames 2004) Available at http://www.public.iastate.edu/~splida

23. Statistical Approaches to Planning of Accelerated Reliability Testing

This chapter presents a few statistical methods for designing test plans in which products are tested under harsher environment with more severe stresses than usual operating conditions. Following a short introduction, three different types of testing conditions are dealt with in Sects. 23.2, 23.3, and 23.4; namely, life testing under constant stress, life testing in which stresses are increased in steps, and accelerated testing by monitoring degradation data. Brief literature surveys of the work done in these areas precede presentations of methodologies in each of these sections.

In Sect. 23.2, we present the conventional framework for designing accelerated test plans using asymptotic variance of maximum likelihood estimators (MLE) derived from the Fisher information matrix. We then give two possible extensions from the framework for accelerated life testing under three different constant stress levels; one based on a nonlinear programming (NLP) formulation so that experimenters can specify the desired number of failures, and one based on an enlarged solution space so that the design of the test plan can be more flexible in view of the many possible limitations in practice. These ideas are illustrated using numerical examples and followed by a comparison across different test plans.

We then present planning of accelerated life testing (ALT) in which stresses are increased in steps and held constant for some time before the next increment. The design strategy is based on a target acceleration factor which specifies the desired time compression needed to complete the test compared to testing under use conditions.

Using a scheme similar to backward induction in dynamic programming, an algorithm for planning multiple-step step-stress ALT is presented.

In Sect. 23.4, we consider planning problems for accelerated degradation test (ADT) in which degradation data, instead of lifetime data, are used to predict a product's reliability. We give a unifying framework for dealing with both constant-stress and step-stress ADT. An NLP model which minimizes cost with precision constraint is formulated so that the tradeoff between getting more data and the cost of conducting the test can be quantified.

Technology and market forces have created generation after generation of highly reliable products. For most products, it is no longer feasible to test products/components in the usual manner at their design conditions as the time needed to obtain sufficient failure information so as to understand the products' behaviors and to quantify their reliability is prohibitively long. In applications where safety and mission success are critical, accelerated reliability testing (ART) is commonly deployed to quantify products' reliability. The basic idea of ART is to achieve time compression so that failure information can be precipitated within a reasonable

test duration. To achieve the requisite time compression, products are tested either under a harsher environment or under more intensive usage than the usual use condition. For many products which are always on, only the former is feasible and will be dealt with in this chapter.

One of the important issues in ART is how to plan a test by determining the testing time, stress levels, sample sizes for different stress levels, etc., so as to achieve cost saving and/or efficiency. Good planning does not only lead to shorter test time or fewer test specimens or both; but more importantly; a good test plan will result in a more precise estimate for the reliability measure of interest, which could be critical, in some cases, in meeting a design or customer specification.

In this chapter, we present test plans for three types of acceleration tests, namely, constant-stress accelerated life test, step-stress accelerated life test and accelerated degradation test, in the following three sections, respectively. A quick review of the literature in each of these areas will first be presented at the beginning of the respective sections. The statistical approach for planning these experiments will then follow. At the end of each section, a numerical example will be given to illustrate the application of the planning methodology.

23.1 Planning Constant-Stress Accelerated Life Tests

The most commonly adopted ART is the constant-stress accelerated life test (CSALT) which comprises multiple sub-samples tested at different but fixed stress levels, at which time-censored failure times are recorded. For ease of administration, commonly used CSALT test plans consist of equally spaced test stresses, each with the same number of test specimens. Such standard plans are highly inefficient for estimating product reliability at the design stress as fewer failures are expected at lower stress levels unless the test duration is much longer than other higher stress levels. Pioneering work by *Chernoff* [23.1] and *Meeker* and *Nelson* [23.2] proposed statistically optimal plans for constant-stress ALT which involved only two stress levels. As a result, these plans, though statistically optimal, cannot be used to validate the assumed stress–life relationship. To remedy this problem, *Meeker* and *Hahn* [23.3] proposed the use of a 4:2:1 allocation ratio for low-, middle- and high-stress levels and gave the optimal low-stress level by assuming that the middle stress is the average of the high- and low-stress levels. An alternative approach is to set equally spaced stress levels with equal allocations as in *Nelson* and *Kielspinski* [23.4] and *Nelson* and *Meeker* [23.5]. The main reason for using a predetermined allocation and middle stress is that the statistically optimal plan only involves two stress levels once the stress–life relation is given. *Nelson* [23.6] provides a comprehensive treatment of these plans and thus they will not be presented in this chapter.

Another approach adopted by *Yang* [23.7] and *Yang* and *Jin* [23.8] to overcome this problem is to add test constraints involving expected minimum number of failures to be observed at the mid-stress levels. The resulting plan is sensitive to the value of the constraint for expected minimum number of failures as it indirectly determines the mid-stress levels and corresponding sample allocation. *Tang* [23.9] imposed constraints that limit the probability of having probability plots with best-fit lines crossing at the lower tails. In this way, the stress levels and their corresponding sample allocations are best suited to infer whether the shape parameters are indeed different at different stress levels. Motivated by the fact that the mid-stress level is meant for validating the assumed stress–life model, *Tang* et al. [23.10] considered test plans in which the mid-stress level will have the least influence on the slope of the stress–life relationship plot. Although these plans are more robust, variances of the estimates at design stress for these compromised plans may be much higher than those under the two-stress-level optimum plans. To address this, *Tang* and *Yang* [23.11] proposed a graphical approach for planning multiple constant-stress-level ALT so that the uncertainty involved for some estimate of interest is not worse than that of a statistical optimum plan by a margin determined by the experimenter before the test. Recently, *Tang* and *Xu* [23.12] presented a general framework for planning ALT. We shall summarize these plans in this Section and provide a simple comparison.

There are many other papers in statistical planning of CSALT that are not presented here. Most of the work has been summarized in *Nelson* [23.6] and *Meeker* and *Escobar* [23.13]. A comprehensive bibliography is given in *Nelson* [23.14]. For example, *Meeter* and *Meekers* [23.15] presented optimal plans involving a nonconstant shape parameter, in the case of Weibull distribution, and *Tang* et al. [23.16] gave optimal plans involving a failure-free life represented by a location parameter.

Notations and Assumptions

S_d, S_i, S_h	Stress (design, i, high) levels, $i = 1, 2, \ldots, h-1$
ξ_i	$\frac{S_i - S_h}{S_d - S_h}$: normalized stress levels, $i = 1, 2, \ldots, h$, so that $\xi_h = 0, \xi_d = 1$
π_i	Proportion of units tested at ξ_i
n	Total sample size
n_i	Actual number of test units needed at ξ_i
h	Total number of stress levels; also denotes the high stress level
τ_i	Censoring time at low S_i, $i = 1, 2, \ldots, h$
$\mathbf{F}$	Fisher information matrix for the plan with a unit of n independent observations
σ	Scale parameter
μ	Location parameter of the smallest extreme value distribution
μ_D, μ_H	μ (design, highest test stress)
p_i	Probability that units tested at ξ_i fail before τ_i
m	A maximum bound
γ_i	Expected number of failure at stress ξ_i
Avar[log(t)]	Asymptotic variance of natural logarithm of the estimate of a fixed percentile of interest t
$\hat{t}_{0.43}$	The estimate of the 43rd percentile of the time to failure; which is also the maximum likelihood estimation (MLE) of the mean life of the smallest extreme value distribution at design stress

23.1.1 The Common Framework

For ease of discussion and comparison, the common assumptions and framework of these plans are:

1. The lifetime follows a Weibull distribution of which the natural logarithm of life, $y = \ln(t)$, has a smallest extreme-value distribution with a reliability function given by
$$R(y) = \exp\left\{-\exp\left[(y-\mu)\Big/\sigma\right]\right\} \quad -\infty < y < +\infty\,, \tag{23.1}$$
where μ is the location parameter and σ is the scale parameter.
2. The scale parameter does not depend on the stress level and;
3. The location parameter is a linear function of the transformed stress:
$$\mu(\xi) = \alpha + \beta\,\xi\,, \tag{23.2}$$
where, α, β are unknown parameters to be estimated from test data.
4. For planning purpose, the initial guess values for σ, p_D and p_H are given.
5. Test units allocated to ξ_i are tested simultaneously until τ_i.
6. The highest test stress is specified.
7. The test stresses are above the design stress.
8. The measure of statistical precision is given by the asymptotic variance of the MLE of the mean life of the smallest extreme value distribution, which is the 43rd percentile ($\hat{t}_{0.43}$), at design stress.

For test conducted at three stress levels, the above variance is given by:

$$\begin{aligned} y_{\text{var}} &= \text{Avar}\left[\hat{t}_{0.43}(1)\right] \\ &= [1\,,1\,,-0.577\,22]\mathbf{F}^{-1}[1\,,1\,,-0.577\,22]'\,, \end{aligned} \tag{23.3}$$

where $\mathbf{F}$ is the Fisher information matrix for the plan given below

$$\mathbf{F} = \frac{n}{\sigma^2}\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}\,,$$

$$M_{11} = \sum_{i=1}^{3} \pi_i A(\psi_i)\,,$$

$$M_{22} = \sum_{i=1}^{3} \pi_i \xi_i^2 A(\psi_i)\,,$$

$$M_{33} = \sum_{i=1}^{3} \pi_i C(\psi_i)\,,$$

$$M_{12} = M_{21} = \sum_{i=1}^{3} \pi_i \xi_i A(\psi_i)\,,$$

$$M_{13} = M_{31} = \sum_{i=1}^{3} \pi_i B(\psi_i)\,,$$

$$M_{32} = M_{23} = \sum_{i=1}^{3} \pi_i \xi_i B(\psi_i)\,,$$

$$A(\psi_i) = 1 - \exp\left\{-\exp\left[(\psi_i - \theta)\Big/\sigma\right]\right\}\,,$$

$$
\begin{aligned}
B(\psi_i) = & \int_0^{\exp(\psi_i)} \ln(w) w \exp(-w)\,\mathrm{d}w \\
& + \left(\exp\left\{-\exp\left[(\psi_i-\theta)\big/\sigma\right]\right\}\right) \\
& \times \psi_i \exp(\psi_i)\,, \\
C(\psi_i) = & 1-\exp\left\{-\exp\left[(\psi_i-\theta)\big/\sigma\right]\right\} \\
& + \int_0^{\exp(\psi_i)} \ln^2(w) w \exp(-w)\,\mathrm{d}w \\
& + \left(\exp\left\{-\exp\left[(\psi_i-\theta)\big/\sigma\right]\right\}\right) \\
& \times \psi_i^2 \exp(\psi_i)\,, \\
\psi_i = & \left[\tau_i - (\alpha+\beta S_H)\right]\big/\sigma\left[\beta\,(S_D - S_H)\right]\big/\sigma\,.
\end{aligned}
$$

Note that, usually, the standardized asymptotic variance, $\mathrm{Svar} = \mathrm{Avar}^*(n/\sigma^2)$ is used in optimization.

23.1.2 Yang's Approach

Yang [23.7] formulated a constrained nonlinear programme to obtain an optimal test plan. He allowed for varying censoring time at different stress levels and aimed to minimize the total test duration while achieving statistical efficiency and robustness. The objective function to be minimized consists of a weighted sum of the standardized asymptotic variance and the product of the test duration at the lowest stress level and the sum of test duration; i. e.

$$
\begin{aligned}
&\text{Minimize: } \omega\,\mathrm{AVar}\left[\log(t_{0.43})\right] + (1-\omega)\tau_1 \sum_i \tau_i\,, \\
&\qquad 0 \le \omega \le 1\,, \\
&\text{Subject to: } \tau_i > \tau_j \quad \text{for } i < j\,, \\
&\qquad \sum_i \pi_i = 1\,; \\
&\qquad \xi_i > \; xi_j > \xi_h = 0 \quad \text{for } i < j\,, \\
&\qquad n\pi_i p_i \ge \gamma_i \quad \text{for all } i\,.
\end{aligned}
\tag{23.4}
$$

Yang [23.7] presented the solution for four stress levels, i. e. $h = 4$, and $\gamma_i = 10$ for all i. He transformed the above into an unconstrained optimization using a penalty-function approach to obtain nearly optimal plans. A similar approach is given in *Yang* and *Jin* [23.8] with 3 stress levels, known as the three-level best-compromised test plan, where the middle stress is the average of the low and high stresses.

23.1.3 Flexible Near-Optimal Plans

In the above plans, the inclusion of additional stress levels will give rise to less-precise estimates compared to the statistically optimal two-level CSALT. For better management of risk in planning ALT, it would be useful to know the extent by which the variance of the percentile of interest is inflated. For practical implementation, test plans should also be sufficiently flexible, in that some range of stress levels and the corresponding allocations are provided instead of stipulating specific stress levels and allocations. In the following, we present flexible near-optimal plans proposed by *Tang* and *Yang* [23.11] that provides the flexibility while limiting the loss in precision. For simplicity, we assume that the censoring time at all stress levels are identical, i. e. $\tau_i = T$ for all i.

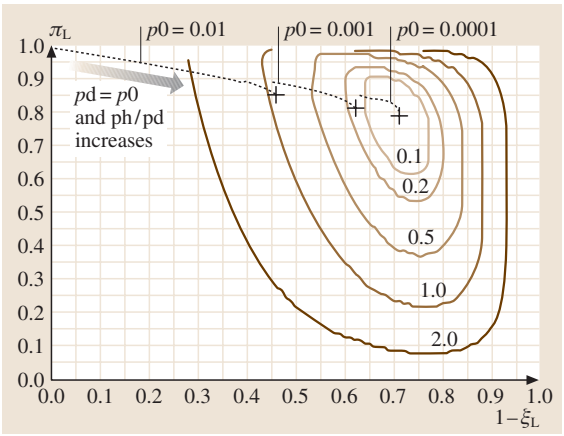


Fig. 23.1 Solution space of a two-stress-level CSALT

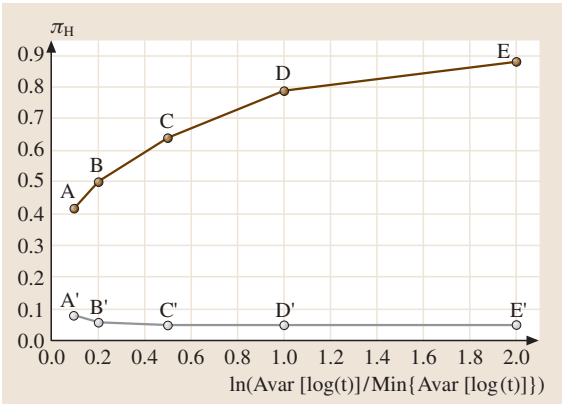


Fig. 23.2 Feasible region of π_H for different values of m in (23.5)

Consider a solution of two-stress-level CSALT depicted in Fig. 23.1. The statistically optimal solutions specify the low stress level (ξ_L) and the corresponding allocation (π_L), which are marked by "+" for different p_d. A possible approach in enlarging the solution space is to consider combinations of (ξ_L, π_L) such that the following ratio is restricted to a maximum bound tolerable, say m:

$$\ln\left(\frac{\mathrm{Avar}\left[\log(t)\right]}{\mathrm{Min}\left\{\mathrm{Avar}\left[\log(t)\right]\right\}}\right) \le m\ . \tag{23.5}$$

Figure 23.1 depicts the contours that enclosed (ξ_L, π_L) for m ranging from 0.1–2.

This principle of enlarging the solution space can be generalized to 3SCSALT. In the following, we give a step-by-step description on how a contour plot of the solution space for 3SCSALT is constructed.

1. For different combinations of p_d and p_h, the statistically optimal 2SCSALT plan with the optimal value of π_H can be determined. By allowing the value of m in (23.5) to vary, a range of π_H as a function of m can easily be determined. The results are depicted in Fig. 23.2 for m= 0.1, 0.2, 0.5, 1.0, 2.0.
2. For each value of π_H, the solution space of ξ_M and ξ_L can be obtained using the same criterion as in (23.5). An example for $\pi_H = 0.15$ is given in Fig. 23.3 for m= 0.1, 0.2, 0.5, 1.0, 2.0. Interestingly, the solution space of ξ_M and ξ_L is enclosed in an approximately right-angled triangle sharing the common slope of 1 as $\xi_L > \xi_M$.
3. As one would usually like to ensure that ξ_L and ξ_M are sufficiently far apart, the preferred solution will be at the vertex of the right angle. Tracing the vertices for different values of m, the various combinations of ξ_L and ξ_M form a straight line, as depicted in Fig. 23.2.
4. Repeated applications of this procedure for different values of π_H result in a plot depicting the solution space of ξ_L and ξ_M of 3SCSALT, as shown in Fig. 23.4. In Fig. 23.4, we use the boundary values of π_H marked by A, B, C, D, E, A', B', C', D' and E' in Fig. 23.2, and some intermediate values of π_H (0.13, 0.18, 0.25, 0.33) to generate the corresponding lines that give the preferred solutions of ξ_L and ξ_M for different values of m.
5. The contours of various setting of m (0.1, 0.2, 0.5, 1, 2) are superimposed on these lines so that ξ_L and ξ_M can be read off by interpolation between the lines of π_H for a given m value.

To determine π_L and π_M, since the main purpose of having a middle stress is to validate the stress life model, one may prefer minimum allocation to the middle stress such that there are sufficient failures to detect nonlinearity, if it exists. In this case, π_M is given by

$$\pi_M = \frac{\gamma_m}{n F_M(T)}\ , \tag{23.6}$$

where γ_M is the minimum number of failures expected under the middle stress level, and $F_M(T)$ is the prob-

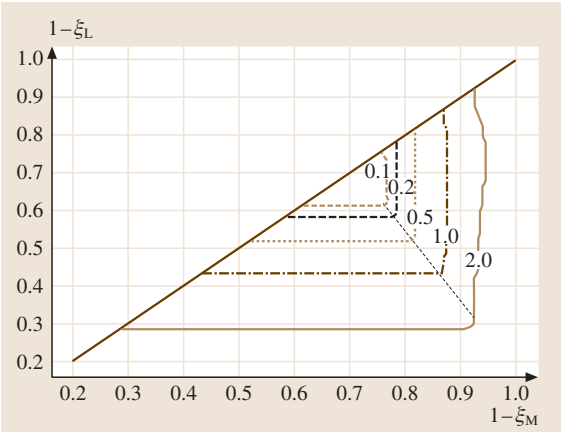


Fig. 23.3 Solution space for the middle and low stress levels for $\pi_H = 0.15$

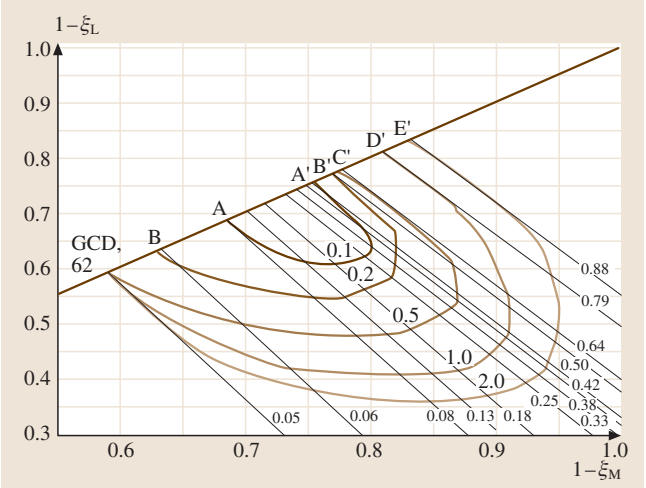


Fig. 23.4 Contour plots of the solution space for 3SCSALT

ability of failure by the end of the test at the middle stress.

Alternatively, as suggested by *Tang* [23.9], a good planning strategy is to set the centroid of the lower and middle stress levels, weighted by their respective allocation, in a near-optimal 3SCSALT plan equal to the optimal low stress in the statistical optimal plan, i. e.

$$\frac{\xi_{3M}\pi_{3M}+\xi_{3L}\pi_{3L}}{\pi_{3M}+\pi_{3L}}=\xi_{2L}^{*}$$

$$\Rightarrow \xi_{3M}\pi_{3M}+\xi_{3L}\pi_{3L}$$

$$=\xi_{2L}^{*}\left(1-\pi_{H}^{*}\right), \qquad (23.7)$$

where the subscript numbers 3 and 2 represent the number of stresses under which the test plans are designed, superscript "*" means optimal values; note that $\pi_{3H}=\pi_{H}^{*}$.

In summary, the steps in obtaining a 3SCSALT test plan are:

- For the given inputs, solve for the optimal π_{H}^{*}.
- For a given value of m, obtain ξ_{L} and ξ_{M} from Fig. 23.4 by interpolation between the lines of π_{H} to that corresponds to the optimal π_{H}.
- Compute π_{M} from (23.6) or (23.7) noting that $\pi_{L}=1-\pi_{M}-\pi_{H}$.

23.1.4 Numerical Example

In this section, we give an example to show the procedure for planning a three-stress-level CSALT given that $p_d=0.0001$, $p_h=0.9$ and $n=300$, $T=300$, $\sigma=1$, and $m=0.1$. These give rise to $\mu_H=4.8698$ and $\mu_D=14.9148$.

1. Solve for the statistically optimal plan: $\xi_{2L^*}=0.29$ and $\pi_H=0.21$. (see Fig. 23.1)
2. From Fig. 23.4, with $\pi_H=0.21$ and $m=0.1$, $\xi_L=0.38$ and $\xi_M=0.22$.
3. From (23.6), since $\mu(\xi_M)=7.0795$ and $F(300)=0.223$, assuming that $\gamma_M=21$, we have $\pi_M=0.314$ and $\pi_L=0.477$.
4. Alternatively, from (23.7), we have $0.22(0.79-\pi_L)+0.38\pi_L=0.29\times0.79$ which gives $\Rightarrow \pi_L=0.35$, $\pi_M=0.44$.

Since the low stress levels are determined with $m<0.1$, the resulting asymptotic variances should be less than 1.1 times the best achievable variance. Note that the sample allocation ratio is approximately 5:3:2, which is quite different from the 4:2:1 recommended by *Meeker* and *Hahn* [23.3]. Despite the lower allocation to the low stress level, the expected number of failures is about 8–10 at the lower stress in these plans, which is sufficient to make some meaningful statistical inference.

23.2 Planning Step-Stress ALT (SSALT)

The simplest form of SSALT is a partially ALT considered by *Degroot* and *Goel* [23.23] in which products are first tested under use condition for a period of time before the stress is increased and maintained at a level throughout the test. This is a special case of a simple SSALT where only two stress levels are used. Much work has been done in this area and literature appeared before 1989 has been covered by *Nelson* [23.6], which has a chapter devoted to step-stress and progressive-stress ALT assuming exponential failure time. A survey

Table 23.1 A summary of the characteristics of literature on optimal design of SSALT

Paper	Problem addressed	Input	Output	Lifetime distribution
Bai et al. [23.17]	Planning two-step SSALT	p_d, p_h	Optimal hold time	Exponential
Bai, *Kim* [23.18]	Planning two-step SSALT	p_d, p_h, shape parameter	Optimal hold time	Weibull
Khamis, *Higgins* [23.19]	Planning three-step SSALT without censoring	All parameters of stress–life relation	Optimal hold time	Exponential
Khamis [23.20]	Planning m-step SSALT without censoring	All parameters of stress–life relation	Optimal hold time	Exponential
Yeo, *Tang* [23.21]	Planning m-step SSALT	p_h and a target acceleration factor	Optimal hold time and lower stress	Exponential
Park, *Yum* [23.22]	Planning two-step SSALT with ramp rate	p_d, p_h, ramp rate	Optimal hold time	Exponential

of the subsequent work from 1989 has been given in *Tang* [23.24]. A summary of the work relating to optimal design of SSALT is presented in Table 23.1. The term optimal usually refers to minimizing the asymptotic variance of the log(MTTF), where MTTF is the mean time to failure, or that of a percentile at use condition. As we can see from Table 23.1, with the exception of *Bai* and *Kim* [23.18], all this work deals with exponential failure time. This is due to simplicity as well as practicality, as it is hard to know the shape parameter of the Weibull distribution in advance.

23.2.1 Planning a Simple SSALT

We first consider a two-level SSALT in which n test units are initially placed on S_1. The stress is changed to S_2 at $\tau_1 = \tau$, after which the test is continued until all units fail or until a predetermined censoring time T. For simplicity, we assume that, at each stress level, the life distribution of the test units is exponential with mean θ_i, and that the linear cumulative exposure model (LCEM) of *Nelson* [23.6] applies.

The typical design problem for a two-step SSALT is to determine the optimal hold time, with a given low stress level. In the following, we shall give the optimal plan that includes both optimal low stress and hold time as in *Yeo* and *Tang* [23.21].

The Likelihood Function

Under exponential failure time and LCEM assumptions, the likelihood function under simple step-stress is

$$L(\theta_1, \theta_2) = \prod_{j=1}^{n_1}\left[\frac{1}{\theta_1}\exp\left(-\frac{t_{1,j}}{\theta_1}\right)\right] \times \prod_{j=1}^{n_2}\left[\frac{1}{\theta_2}\exp\left(-\frac{t_{2,j}-\tau}{\theta_2}-\frac{\tau}{\theta_1}\right)\right] \times \prod_{j=1}^{n_c}\exp\left(-\frac{\tau}{\theta_1}-\frac{T-\tau}{\theta_2}\right), \quad (23.8)$$

where the notations are defined as follows:

- n_i number of failed units at stress level S_i, $i = 1, 2, \ldots, h$,
- n_c number of censored units at S_h (at end of test),
- $t_{i,j}$ failure time j of test units at stress level S_i, $i = 1, 2, \ldots, h$,
- θ_i mean life at stress S_i, $i = 1, 2, \ldots, h$,
- τ_i hold time at low stress levels S_i, $i = 1, 2, \ldots, h-1$,
- T censoring time.

MLE and Asymptotic Variance

The MLE of $\log(\theta_0)$ can be obtained by differentiating the log-likelihood function in (23.8):

$$\log\left(\theta_0\hat{}\right) = \frac{\log\left(\frac{U_1}{n_1}\right) - (1-\xi_1)\log\left(\frac{U_2}{n_2}\right)}{\xi_1}, \quad (23.9)$$

where

$$U_1 = \sum_{j=1}^{n_1} t_{1,j} + (n-n_1)\cdot\tau\,;$$

and

$$U_2 = \sum_{j=1}^{n_2}\left(t_{2,j}-\tau\right) + (n-n_c)\cdot(T-\tau)\,.$$

From the Fisher information matrix, the asymptotic variance of the MLE of the log(mean life) at the design stress is:

$$V(\xi_1, \tau) = \frac{\left(\frac{1}{\xi_1}\right)^2}{1-\exp\left(-\frac{\tau}{\theta_1}\right)} + \frac{\left(\frac{1-\xi_1}{\xi_1}\right)^2}{\exp\left(-\frac{\tau}{\theta_1}\right)\left[1-\exp\left(-\frac{1-\tau}{\theta_2}\right)\right]}\,. \quad (23.10)$$

To obtain the optimal test plan, we need to express (23.10) in terms of ξ_1,τ, and other input variables. To do this, we need to assume a stress–life relation. For illustration, suppose the mean life of a test unit is a log-linear function of stress:

$$\log(\theta_i) = \alpha + \beta S_i\,, \quad (23.11)$$

where α, β ($\beta < 0$) are unknown parameters. (This is a common choice for the life–stress relationship because it includes both the power-law and the Arrhenius relation as special cases.) From the log-linear relation of the mean in (23.11), we have

$$\frac{\theta_2}{\theta_1} = \left(\frac{\theta_2}{\theta_0}\right)^{\xi_1} = \exp(\beta\xi_1)\,.$$

And since $p_h = 1-\exp\left(-\frac{1}{\theta_2}\right)$, it follows that $V(\xi, \tau)$ is given by:

$$V(\xi_1, \tau) = \frac{\left(\frac{1}{\xi_1}\right)^2}{1-(1-p_h)^{\omega}} + \frac{\left(\frac{1-\xi_1}{\xi_1}\right)^2}{(1-p_h)^{\omega}\left[1-(1-p_h)^{1-\tau}\right]}\,, \quad (23.12)$$

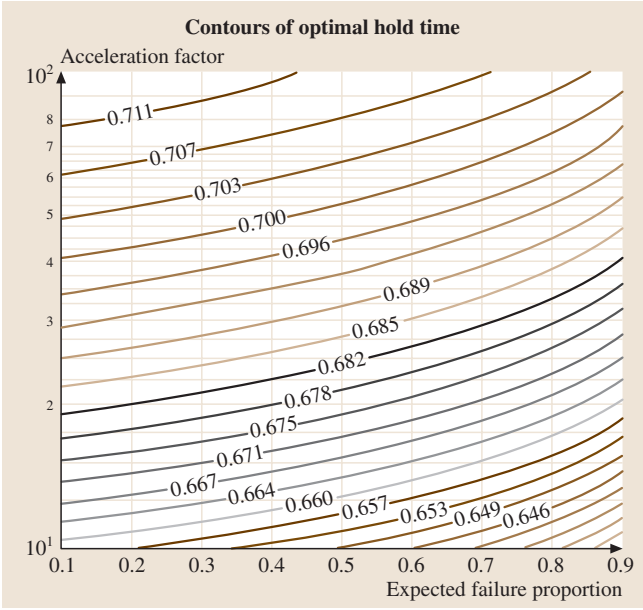


Fig. 23.5 Contours of optimal hold time at low stress for a two-step SSALT. For a given (p, ϕ), the optimal hold time can be read off by interpolation between the contours

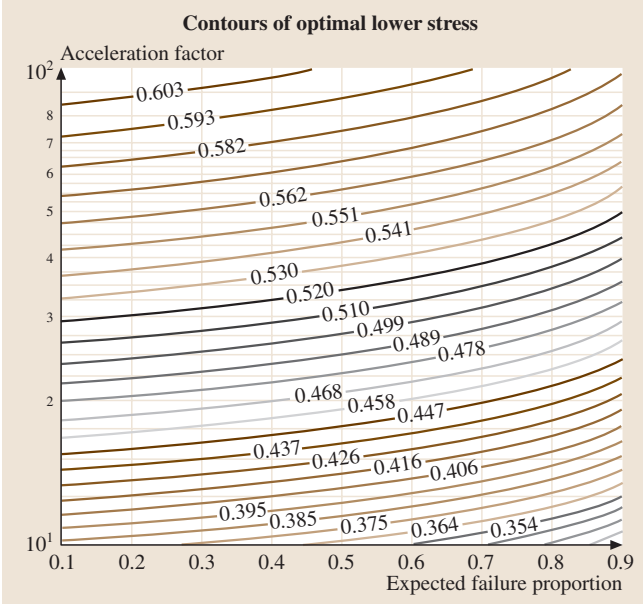


Fig. 23.6 Contours of optimal low stress level for a two-step SSALT. For a given (p, ϕ), the optimal low stress can be read off by interpolation between the contours

where

$$\omega \equiv \tau \left(\frac{\theta_2}{\theta_0}\right)^{\xi_1} = \tau \left[\exp\left(\beta\xi_1\right)\right].$$

As β is unknown, another input variable is needed. We propose using the target acceleration factor (AF) as it is a measure of the amount of extrapolation and a time-compression factor. AF is easier to estimate compared to the commonly used probability of failure at design stress. Given the time constraint that determines the maximum test duration and some guess of the test duration if the test were conducted at use condition, the target AF, denoted by ϕ, is given by the ratio of the two, i. e.

$$\phi = \frac{\text{time to failure at design stress}}{\text{time to failure under test plan}} . \tag{23.13}$$

For exponential lifetime under the LCEM assumption, the equivalent operating time for the test duration T at the design stress given is by $\theta_0 \left(\frac{\tau}{\theta_1} + \frac{T-\tau}{\theta_2}\right)$. As a result, the AF is:

$$\phi = \frac{\tau\frac{\theta_0}{\theta_1} + (T-\tau)\frac{\theta_0}{\theta_2}}{T} . \tag{23.14}$$

From the log-linear stress–life relation in (23.11), without loss of generality, let $S_0 = 0$, $S_1 = x$, $S_2 = 1$, $T = 1$. Then, (23.14) becomes:

$$\phi = (1-\tau)\exp\left(-\beta\right) + \tau\exp\left(-\beta x\right) . \tag{23.15}$$

The optimal low stress and the corresponding hold time can be obtained by solving the following constrained nonlinear programme (NLP):

$$\text{min: } V(x,\tau) = \frac{\left(\frac{1}{1-x}\right)^2}{1-(1-p_h)^{\omega}} + \frac{\left(\frac{x}{1-x}\right)^2}{(1-p_h)^{\omega}\left[1-(1-p_h)^{1-\tau}\right]}$$

$$\text{subject to: } (1-\tau)\exp\left(-\beta\right) + \tau\exp\left(-\beta x\right) = \phi , \tag{23.16}$$

where $x = 1-\xi_1$.

The results are given graphically in Figs. 23.5, 23.6, with (p_h, ϕ) on the x–y axis, for ϕ ranging from 10–100 and p ranging from 0.1–0.9. Figure 23.5 shows the contours of the optimal normalized hold time τ and Fig. 23.6 gives the contours of the optimal normalized low stress (x). Given a pair of (p_h, ϕ) the simultaneous

optimal low stress and hold time can be read from the graphs.

Both sets of contours for the optimal hold time and the optimal low stress show an upward convex trend. The results can be interpreted as follows. For the same p, in situations where the time constraint calls for a higher AF, it is better to increase the low stress and extend the hold time at low stress. For a fixed AF, a longer test time (so that p increases) will result in a smaller optimal low stress and a shorter optimal hold time.

23.2.2 Planning Multiple-Step SSALT

To design optimal plans for multiple-step SSALT, we adopt a similar idea as in CSALT where the low stress level and its sample allocation are split into two portions. As all units are tested in a single step-stress pattern, the analogy of the sample allocation in a CSALT is the hold time in a SSALT. As a result, for a three-step SSALT, the hold time at the high stress level is kept at $(1-\tau)$ while the hold time at low stress is split into two for the additional intermediate stress level. In doing so, we need to ensure that after splitting the optimal low stress of a two-step SSALT into two stress levels, the AF achieved in the newly created three-step SSALT is identical to that the original two-step SSALT. Since the high stress and its hold time remain intact, the AF contributed by the first two steps of the three-step SSALT must be the same as that contributed by the low stress in the original two-step SSALT. In essence, given the target AF and p_h, we first solve the optimal design problem for a two-step SSALT. Then, a new target AF corresponding to the AF contributed by the low stress of the optimal two-step SSALT is used as input to solve for a new two-step SSALT; which, after combining with the earlier result, forms a three-step SSALT. To achieve the new target AF, the resulting middle stress will be slightly higher than the optimal low stress. The new two-step SSALT uses this middle stress as the high stress and the optimal τ as the test duration. The optimal hold time and low stress for a three-step SSALT can be solved using (23.16).

The above procedure can be generalized to m steps SSALT which has $m-1$ cascading stages of two-step SSALT, as it has the structure of a typical dynamic programme. The number of steps corresponds to the stage of a dynamic programme and the alternatives at each stage are the low stress level and test duration.

For example, in a three-step SSALT having two cascading stages of simple SSALT, the results of stage 1 of the simple SSALT gives the optimal low stress level and its hold time. At stage 2, this low stress level is split into a simple SSALT that maintains the overall target AF. As a result AF is one of the state variables which is additive under LCEM; i. e. the new target AF for stage 2 is the AF contributed by the low stress in stage 1. From (23.15), the AF contributed by the low stress is given by

$$\phi' = \tau \ \exp(-\beta x)\,. \tag{23.17}$$

To solve the stage 2 NLP, this target AF needs to be normalized by the hold time, τ, due to change to time scale; i. e. the normalized AF, ϕ_2 at stage 2 is given by:

$$\phi_2 = \phi'/\tau = \exp(-\beta x)\,. \tag{23.18}$$

Note that, to meet the above target AF, the middle stress level x_m should be higher than the optimal low stress in stage 1. As a result, p_2, the expected proportion of failure at x_m during τ will also increase. At the same time, for consistency, we need to ensure that β of the stress life model is identical to that obtained in stage 1. These variables are interdependent and can only be obtained iteratively. The algorithm [23.24] that iteratively solves for p_2, x_m, that results in the same β is summarized as follows.

1. Compute $p_2^{(0)}$, the expected proportion of failure at the low stress level of stage 1:

$$\begin{aligned} p_2^{(0)} &= 1-\exp\left(-\frac{\tau}{\theta_x}\right) \\ &= 1-\exp\left\{-\tau\log\left(\frac{1}{1-p}\right)\right. \\ &\quad \left.\times\exp\left[\beta\,(1-x)\right]\right\}. \end{aligned} \tag{23.19}$$

2. Solve the constrained NLP in (23.16) to obtain $\left(\tau^*_{(1)}, x^*_{(1)}, \beta^*_{(1)}\right)$.
3. Compute the new middle stress $x_\mathrm{m}^{(1)}$

$$x_\mathrm{m}^{(1)} = \beta^*_{(1)} \Bigg/ \left[\log\left(\frac{\log\left(\frac{1}{1-p_2^{(0)}}\right)}{\tau\log\left(\frac{1}{1-p}\right)}\right) + \beta^*_{(1)}\right], \tag{23.20}$$

4. Update p_2 using $x_\mathrm{m}^{(1)}$:

$$\begin{aligned} p_2^{(1)} &= 1-\exp\left\{-\tau\log\left(\frac{1}{1-p}\right)\right. \\ &\quad \left.\times\exp\left[\beta\left(1-x_\mathrm{m}^{(1)}\right)\right]\right\}. \end{aligned} \tag{23.21}$$

5. Repeat steps 2 to 4, with $\left(p_2^{(1)}, \phi_2\right)$, $\left(p_2^{(2)}, \phi_2\right)$, $\left(p_2^{(3)}, \phi_2\right), \ldots$ until $\left|\beta^*_{(k)} - x_\mathrm{m}^{(k)}\beta\right| < \varepsilon$, for some prespecified $\varepsilon > 0$.

Suppose that $\left(\tau^*_{(k)}, x^*_{(k)}, \beta^*_{(k)}\right)$ are the solutions of the scheme. The new optimal low stress and hold time can be computed by combining the results from stages 1 and 2 to form the optimal plan for a three-step SSALT ($h = 3$) as follows:

$$x_1 = x_{\mathrm{m}}^{(k)} \cdot x^*_{(k)}\ ,\ \ x_2 = x_{\mathrm{m}}^{(k)}\ ,\ \ \tau_1 = \tau \cdot \tau^*_{(k)}\ ,\ \ \tau_2 = \tau\ . \tag{23.22}$$

In general, the above scheme can be carried out recursively to generate test plans for multiple-step SSALT.

23.2.3 Numerical Example

Suppose that a three-step SSALT test plan is needed to evaluate the breakdown time of insulating oil. The high and use stress levels are 50 kV and 20 kV, respectively, and the stress is in log(kV). The total test time is 20 h. It is estimated that the reliability is about 0.1 under 50 kV for 20 h ($p = 0.9$) and the target acceleration factor $\phi = 50$.

Solving (23.16) gives $\beta = 4.8823$, $\tau_2 = 0.6871$ and $x = 0.5203$. Using these as the inputs for the next stage, we have $\tau_1 = 0.4567$; $x_1 = 0.2865$ and $x_2 = 0.6952$. As a result, the voltages for conducting the SSALT are

$$\begin{aligned} S_1 &= \exp\left[S_0 + x_1(S_h - S_0)\right] \\ &= \exp\left\{\log(20) + 0.2865 \times \left[\log(50) - \log(20)\right]\right\} \\ &= 26.0\,\mathrm{kV}\ , \\ S_2 &= \exp\left[S_0 + x_2(S_h - S_0)\right] \\ &= \exp\left\{\log(20) + 0.6952 \times \left[\log(50) - \log(20)\right]\right\} \\ &= 37.8\,\mathrm{kV}\ . \end{aligned}$$

And the switching times for the lowest and the middle stress are

$$\begin{aligned} t_1 &= \tau_1 20 = 9.134\,\mathrm{h} = 548.0\,\mathrm{min}\ ; \\ t_2 &= \tau_2 20 = 13.74\,\mathrm{h} = 824.5\,\mathrm{min}\ . \end{aligned}$$

The three-step SSALT starts the test at 26 kV and holds for 548 min, then increase the stress to 37.8 kV and holds for another 276 min (=824.5–548), after which the stress is increased to 50 kV until the end of the test.

23.3 Planning Accelerated Degradation Tests (ADT)

For reliability testing of ultra-high-reliability products, ALT typically ends up with too few failures for meaningful statistical inferences. To address this issue, accelerated degradation tests (ADT), which eliminate the need to observe actual failures, were proposed. For successful application of ADT, it is imperative to identify a quantitative parameter (degradation measure) that is strongly correlated with product reliability and thus will degrade over time. The degradation path of this parameter is then synonymous to performance loss of the product. The time to failure is usually defined as the first passage time of the degradation measure exceeding a prespecified threshold.

Planning of ADT typically involve specifying the stress levels, sample size, sample allocations, inspection frequencies and number of inspections for a constant-stress ADT (CSADT). More samples and frequent inspections will result in more accurate statistical inferences; but at a higher testing cost. So there is a tradeoff between the attainable precision of the estimate and the total testing cost. *Park* and *Yum* [23.25] and *Yu* and *Chiao* [23.26] used precision constraints for optimal planning. *Boulanger* and *Escobar* [23.27] and *Yu* and *Tseng* [23.28] also derived cost functions according to their test procedures. *Wu* and *Chang* [23.29] and *Yang* and *Yang* [23.30] presented CSADT plans such that the asymptotic variance of a percentile of interest is minimized while the testing cost is kept at a prescribed level. *Tang* et al. [23.31] gave SSADT plans in which the testing cost is minimized while fulfilling a precision constraint. *Park* et al. [23.32] gave an SSADT plan with destructive inspections. *Yu* and *Tseng* [23.33] presented a CSDT plan in which the rate of degradation follows a reciprocal Weibull distribution.

23.3.1 Experimental Set Up and Model Assumptions

For simplicity, we consider an ADT, be it a CSADT or SSADT, with two stress levels. Some descriptions and assumptions are as follows:

1. The test stress X_k is normalized by $X_k = \frac{S_k - S_0}{S_2 - S_0}$, $k = 0, 1, 2$, in which the S_k are functions of the applied stresses. With such a transformation, $X_0 = 0 < X_1 < X_2 = 1$.
2. A total sample size is n of which n_k are assigned to the stress level X_k, so that the relationship between

n and n_k can be expressed by:

$$n = \begin{cases} \sum_{k=1}^{2} n_k & \text{for CSADT} \\ n_k & \text{for SSADT} \end{cases} . \tag{23.23}$$

3. The test duration at stress X_k is τ_k, and the stopping time of the whole test is T. The relationship between T and τ_k is:

$$T = \begin{cases} \text{maximum}(T_1, T_2) & \text{for CSADT} \\ \sum_{k=1}^{2} T_k & \text{for SSADT} \end{cases} . \tag{23.24}$$

4. For each unit i, let $D_{i,1}, D_{i,2}, \ldots D_{i,j}$ be the recorded degradation values, which are the differences between the initial and current value of the degradation measurement at the preset time points $t_{i,1}, t_{i,2}, \ldots t_{i,j}, t_{i,0} = 0 < t_{i,1} < t_{i,2} < \ldots < t_{i,j} < \ldots t_{i,L} = T$; each item is measured L_1 times at X_1 and L_2 times at X_2. The total number of inspections is $L = L_1 + L_2$. Given the number of inspections, all samples are inspected simultaneously at equal interval Δt so that the stress-changing time is $T_1 = L_1 \Delta t$ and the experiment stopping time is $T = L\Delta t = (L_1 + L_2)\,\Delta t$.
5. The degradation is governed by a stochastic process $[D_k(t), t \geq 0]$ with drift $\eta_k > 0$ and diffusion $\sigma_k^2 > 0$ at X_k. We assume that the degradation increments follow a normal distribution, i. e. $\Delta D_{i,j} \sim N\left(\eta_k \Delta t_{i,j}, \sigma^2 \Delta t_{i,j}\right)$ with probability distribution function (PDF)

$$f\left(\Delta D_{i,j}\right) = \frac{1}{\sqrt{2\pi}\sqrt{\Delta t_{i,j}\sigma^2}} \exp \times \left(-\frac{\left(\Delta D_{i,j} - \Delta t_{i,j}\eta_k\right)^2}{2\Delta t_{i,j}\sigma^2} \right) \tag{23.25}$$

in which the drift is stress-dependent, and is given by:

$$\eta_k = a + bX_k \tag{23.26}$$

and the diffusion remains constant for all stresses:

$$\sigma_k^2 = \sigma^2 , \tag{23.27}$$

where a, b and σ^2 are unknown parameters that need to be pre-estimated either from engineering handbooks or other ways before experiment planning.

6. Only degradation increments are measured throughout the test. This assumption is mild since the products in ADT are always highly reliable and normally no physical failures occur.

The above model is applicable to stress-drift relations that can be linearized as in (23.26). For example, for degradation induced by humidity, (23.26) may be the result after taking logarithm of the rate of reaction versus the logarithm of the relative humidity. In the case of the Arrhenius model, the reaction rate is an exponential function of the stress factor (= 1/absolute temperature); taking logarithms on both sides of the equation results in a linear function between the log(Drift), which is η_k, versus the stress factor (= 1/absolute temperature).

23.3.2 Formulation of Optimal SSADT Plans

Here, we follow *Tang* et al. [23.31], in which an optimal SSADT plan is obtained such that the total test expense is minimized while the probability that the estimated mean lifetime at use stress locates within a predescribed range of its true value should not be less than a precision level p. For simplicity, the decision variables are the sample size and the number of inspections at each stress level, which also determine the test duration for a given inspection interval. The context of discussion can be generalized to that of a CSADT plan by noting that

$$\pi_k = \begin{cases} \frac{n_k}{n} & \text{for CSADT} , \\ \frac{T_k}{T} & \text{for SSADT} . \end{cases} \tag{23.28}$$

$$q_k = \begin{cases} \frac{T_k}{T} & \text{for CSADT} , \\ \frac{n_k}{n} & \text{for SSADT} . \end{cases} \tag{23.29}$$

For CSADT, $q_1 = 1, 0 < q_2 \leq 1$, while $q_k = 1$ for SSADT. With this definition, the proportion of sample allocation in CSADT is analogous to the holding time in SSADT. This is consistent with the analogy between CSALT and SSALT.

Precision Constraint in SSADT Planning

Suppose that the mean lifetime at use condition, $\mu(X_0)$, is of interest in our planning. To obtain an estimate close to its true value with a certain level of confidence, we impose a precision constraint by limiting the sampling risk in estimating $\mu(X_0)$ with its MLE, i. e. $\widehat{\mu}(X_0)$, to be reasonably small. Mathematically, this can be expressed as follows:

$$\Pr\left[\frac{\mu\,(X_0)}{c} \leq \widehat{\mu}\,(X_0) \leq c\mu\,(X_0) \right] \geq p\,, \tag{23.30}$$

where $c > 1$ and p are given constants. The asymptotic variance of $\widehat{\mu}\,(X_0)$ is needed for further explanation of (23.30). From (23.25), the log-likelihood for an individual degradation increment $D_{i,j}$ is

$$\ln L_{i,j} = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln\left(\Delta t_{i,j}\right) - \ln\sigma - \frac{U_{i,j}^2}{2}\,, \tag{23.31}$$

where

$$U_{i,j} = \frac{\left(\Delta D_{i,j} - \Delta t_{i,j}\eta_k\right)}{\sqrt{\Delta t_{i,j}}\sigma}\,, \begin{cases} k=1 & \text{if } j \le L_1\,, \\ k=2 & \text{otherwise}\,. \end{cases}$$

Hence, the log-likelihood function for all degradation increments of n items is given by

$$\ln L = \sum_{i=1}^{n}\sum_{j=1}^{L}\ln L_{i,j}\,. \tag{23.32}$$

Given the degradation critical value D_c, $\mu\,(X_0)$ is given by the ratio of this threshold value over drift at use condition:

$$\mu\,(X_0) = D_c/\eta_0 = D_c/a\,. \tag{23.33}$$

Let $\left\{\widehat{a}, \widehat{b}, \widehat{\sigma}\right\}$ be the MLE of $\{a, b, \sigma\}$, then, by the invariant property, the MLE of $\mu\,(X_0)$ is given by:

$$\widehat{\mu}\,(X_0) = D_c/\widehat{\eta_0} = D_c/\widehat{a} \tag{23.34}$$

Then the asymptotic variance of $\widehat{\mu}(X_0)$ can be obtained by:

$$\text{Avar}\left[\widehat{\mu}(X_0)\right] = \widehat{h}'\mathbf{F}^{-1}\widehat{h}\,, \tag{23.35}$$

where $h = \left(\frac{\partial\widehat{\mu}(X_0)}{\partial a}, \frac{\partial\widehat{\mu}(X_0)}{\partial b}, \frac{\partial\widehat{\mu}(X_0)}{\partial\sigma}\right)'$, and $\boldsymbol{F}$ is a Fisher information matrix displayed as follows, in which the caret ($\widehat{\ }$) indicates that the derivative is evaluated at $\{a\,, b\,, \sigma\} = \left\{\widehat{a}\,, \widehat{b}\,, \widehat{\sigma}\right\}$. We make use of $E\left(U_{i,j}\right) = 0$ and $\text{Var}\left(U_{i,j}\right) = 1$

$$\boldsymbol{F} = \begin{pmatrix} E\left(-\frac{\partial^2\ln\tilde{L}}{\partial a^2}\right) & E\left(-\frac{\partial^2\ln\tilde{L}}{\partial a\partial b}\right) & E\left(-\frac{\partial^2\ln\tilde{L}}{\partial a\partial\sigma}\right) \\ & E\left(-\frac{\partial^2\ln\tilde{L}}{\partial b^2}\right) & E\left(-\frac{\partial^2\ln\tilde{L}}{\partial b\partial\sigma}\right) \\ \text{symmetrical} & & E\left(-\frac{\partial^2\ln\tilde{L}}{\partial\sigma^2}\right) \end{pmatrix}$$

$$= \frac{n}{\widehat{\sigma}^2}\begin{pmatrix} L\Delta t & \Delta t\sum_{k=1}^{2}X_k L_k & 0 \\ & \Delta t\sum_{k=1}^{2}X_k^2 L_k & 0 \\ \text{symmetrical} & & 2L \end{pmatrix}.$$

Thus we have

$$\text{Avar}\left[\widehat{\mu}(X_0)\right] = \frac{\widehat{\sigma}^2}{n}\frac{D_c^2}{\widehat{a}^4}\left(\frac{\sum_{k=1}^{2}X_k^2 L_k}{L\Delta t\sum_{k=1}^{2}X_k^2 L_k - \Delta t\left(\sum_{k=1}^{2}X_k L_k\right)^2}\right). \tag{23.36}$$

Because the MLE is asymptotically normal and consistent, for large n, approximately we have

$$\widehat{\mu}\,(X_0) \sim N\left\{\left[\mu\,(X_0)\right]\,,\ \text{Avar}\left[\widehat{\mu}\,(X_0)\right]\right\}, \tag{23.37}$$

which can be rewritten as

$$\frac{\widehat{\mu}(X_0)}{\mu(X_0)} \sim N\left(1\,, \frac{\widehat{\sigma}^2}{n\widehat{a}^2}Q\right). \tag{23.38}$$

From (23.30), we have

$$\Pr\left(\frac{1}{c} \le \frac{\widehat{\mu}(X_0)}{\mu(X_0)} \le c\right) \ge p\,. \tag{23.39}$$

This translates into the precision constraint

$$\Phi\left(\frac{(c-1)\sqrt{n}}{\frac{\widehat{\sigma}}{\widehat{a}}\sqrt{Q}}\right) - \Phi\left(\frac{\left(\frac{1}{c}-1\right)\sqrt{n}}{\frac{\widehat{\sigma}}{\widehat{a}}\sqrt{Q}}\right) \ge p\,,\ c > 1\,, \tag{23.40}$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution and

$$Q = \frac{\sum_{k=1}^{2}X_k^2 L_k}{L\times\Delta t\times\sum_{k=1}^{2}X_k^2 L_k - \Delta t\times\left(\sum_{k=1}^{2}X_k L_k\right)^2}. \tag{23.41}$$

Cost Function in SSADT Planning

Typical cost components for testing consists of:

1. Operating cost, which mainly comprises the operator's salary and can be expressed as $\Delta t(C_{o1}L_1 + C_{o2}L_2)$, where C_{ok} is the salary of the operator per unit of time at X_k.
2. Measurement cost, which includes the cost of using measuring equipments and the expense of testing materials. Because depletion of equipments at higher stress is more severe than that at lower stress, measurement cost can be generated as $n(C_{m1}L_1 + C_{m2}L_2)$, where C_{mk} is the cost per measurement per device at X_k.

3. Sample cost, which is related to the number of samples, and can be formulated as $C_d n$, where C_d is the price of an individual device.

So, the total cost (TC) of testing is:

$$\begin{aligned} &TC\,(n\,,\,L_1\,,\,L_2|X_1\,,\,\Delta t) \\ &= \Delta t\,(C_{o1}L_1 + C_{o2}L_2) \\ &\quad + n\,(C_{m1}L_1 + C_{m2}L_2) + C_d n\,, \\ &C_{ok} > 0\,,\, C_{mk} > 0\,,\, C_d > 0\,. \end{aligned} \tag{23.42}$$

In some experiments, the lower test stress can be fixed because of practical limitations. For example, the temperature of a test oven can only be adjusted within a small range or even fixed at some particular values. Given the lower stress X_1, the two-step-stress ADT planning problem is to determine the sample size n, number of inspections L_1 and L_2. The problem is formulated as:

$$\begin{aligned} &\text{Min: TC}\,(n\,,\,L_1\,,\,L_2|X_1\,,\,\Delta t) \\ &\quad = \Delta t\,(C_{o1}L_1 + C_{o2}L_2) \\ &\quad\quad + n\,(C_{m1}L_1 + C_{m2}L_2) + C_d n\,, \\ &\quad\quad C_{ok} > 0\,,\, C_{mk} > 0\,,\, C_d > 0\,; \\ &\text{s.t.: } \Phi\left(\frac{(c-1)\cdot\sqrt{n}}{\frac{\tilde{\sigma}}{\tilde{a}}\cdot\sqrt{Q}}\right) - \Phi\left(\frac{(\frac{1}{c}-1)\cdot\sqrt{n}}{\frac{\tilde{\sigma}}{\tilde{a}}}\cdot\sqrt{Q}\right) \\ &\quad\quad \geq p\,,\;\; c > 1\,. \end{aligned} \tag{23.43}$$

Due to the simplicity of the objective function and the integer restriction on the decision variables, the solution can be obtained by complete enumerations or using search methods given in *Yu* and *Tseng* [23.28].

23.3.3 Numerical Example

In this example, the operating temperature of a light-emitting diode (LED) in use condition is 25 °C. Historical experience indicates that the highest temperature that will not affect the failure mechanism is 65 °C. The lower test stress is set at 45 °C, which can be normalized by

$$\begin{aligned} X_1 &= \frac{S_1 - S_0}{S_2 - S_0} \\ &= \frac{1/(45+273) - 1/(25+273)}{1/(65+273) - 1/(25+273)} \\ &= 0.53\,. \end{aligned}$$

This normalization is consistent with the Arrhenius model in which stress takes the reciprocal of temperature.

To set the inspection time interval, we refer to a similar CSADT plan conducted at 25 °C in *Yu* and *Chiao* [23.26], which suggested an optimal inspection time interval of 240 h. Here, in view of adopting a higher stress, Δt should be shorter as the degradation rate is higher. Here, $\Delta t = 120$ h to capture more degradation information. The operation and measurement coefficients are set at $C_{o1} = 0.3\,,\, C_{o2} = 0.4\,,\, C_{m1} = 4$ and $C_{m2} = 4.5$.

Here the c and p values represent the dependence on sampling risk. Smaller sampling risk implies smaller c and relatively larger p and vise versa. As an illustration, we present the case of $c = 2$, $p = 0.9$ by setting $\tilde{\sigma} = 10^{-4}$ (which is comparable with the value used in *Yu* and *Chiao* [23.26]).

Substitute this information into (23.42) and (23.43), we have:

$$\begin{aligned} &\text{Min:}\quad \text{TC}\,(n\,,\,L_1\,,\,L_2) \\ &\qquad = 120\cdot(0.3\cdot L_1 + 0.4L_2) \\ &\qquad\quad + n\cdot(4L_1 + 4.5L_2) + 86n \\ &\text{s.t.:}\quad \Phi\left(\frac{\sqrt{n}}{100\sqrt{Q}}\right) - \Phi\left(\frac{-\frac{1}{2}\sqrt{n}}{100\sqrt{Q}}\right) \geq 0.9\,, \end{aligned}$$

where $Q = 0.53^2 L_1 + L_2/120\big[(L_1+L_2)(0.53^2 L_1 + L_2) - (0.53L_1 + L_2)^2\big]$.

This plan puts 16 samples at 45 °C for 3240 h, after which the temperature is increased to 65 °C and held for 720 h before the end of the test. Measurements are taken at 120 h interval.

23.4 Conclusions

In this chapter, literature surveys and statistical approaches for planning three types of accelerated reliability testing, namely, constant-stress accelerated life tests, step-stress accelerated life tests and step-stress accelerated degradation tests, are presented. We only focus on literature concerning the above three problems since the 1990s. A more comprehensive survey can be obtained from *Nelson* [23.14]. The general approach taken in solving for the optimal plan is to derive the asymptotic variance (or its approximation) of a percentile of interest at use condition and minimize it subject to a set of constraints. The constraints either

help to define the logical solution space or help to narrow the solution space for ease of finding the solution. For more general considerations, the approach presented in *Tang* and *Xu* [23.12] can be adopted to generalize the current models so that other objectives and constraints can be incorporated.

References

23.1 H. Chernoff: Optimal accelerated life designs for estimation, Technometrics **4**, 381–408 (1962)

23.2 W. Q. Meeker, W. B. Nelson: Optimum censored accelerated life tests for Weibull, extreme value distributions, IEEE Trans. Reliab. **24**, 321–332 (1975)

23.3 W. Q. Meeker, G. J. Hahn: How to plan accelerated life tests: some practical guidelines, ASQC Basic Ref. Qual. Control: Stat. Tech. **10** (1985)

23.4 W. B. Nelson, T. J. Kielspinski: Theory for optimum censored accelerated life tests for normal and lognormal life distributions, Technometrics **18**, 105–114 (1976)

23.5 W. B. Nelson, W. Q. Meeker: Theory for optimum accelerated life tests for Weibull and extreme value distributions, Technometrics **20**, 171–177 (1978)

23.6 W. B. Nelson: *Accelerated Testing: Statistical Models, Test Plans and Data Analysis* (Wiley, New York 1990)

23.7 G. B. Yang: Optimum constant-stress accelerated life-test plans, IEEE Trans. Reliab. **43**, 575–581 (1994)

23.8 G. B. Yang, L. Jin: Best compromise test plans for Weibull distributions with different censoring times, Qual. Reliab. Eng. Int. **10**, 411–415 (1994)

23.9 L. C. Tang: Planning for accelerated life tests, Int. J. Reliab. Qual. Saf. Eng. **6**, 265–275 (1999)

23.10 L. C. Tang, A. P. Tan, S. H. Ong: Planning accelerated life tests with three constant stress levels, Comp. Ind. Eng. **42**, 439–446 (2002)

23.11 L. C. Tang, G. Yang: Planning multiple levels constant stress accelerated life tests, Proc. Ann. Reliab. Maintainab. Symp. , 338–342 (2002)

23.12 L. C. Tang, K. Xu: A multiple objective framework for planning accelerated life tests, IEEE Trans. Reliab. **54**(1), 58–63 (2005)

23.13 W. Q. Meeker, L. A. Escobar: *Statistical Methods for Reliability Data* (Wiley, New York 1998)

23.14 W. B. Nelson: A bibliography of accelerated test plans, Proc. Ninth ISSAT Int. Conf. Reliability and Quality in Design, Honolulu , 189–193 (2003)available from WNconsult@aol.com

23.15 C. Meeter, W. Q. Meeker: Optimum acceleration life tests with a non-constant scale parameter, Technometrics **36**, 71–83 (1994)

23.16 L. C. Tang, T. N. Goh, Y. S. Sun, H. L. Ong: Planning ALT for censored two-parameter exponential distribution, Naval Res. Log. **46**, 169–186 (1999)

23.17 D. S. Bai, M. S. Kim, S. H. Lee: Optimum simple step-stress accelerated life tests with censoring, IEEE Trans. Reliab. **38**, 528–532 (1989)

23.18 D. S. Bai, M. S. Kim: Optimum simple step-stress accelerated life test for Weibull distribution and type I censoring, Naval Res. Log. Q. **40**, 193–210 (1993)

23.19 I. H. Khamis, J. J. Higgins: Optimum 3-step step-stress tests, IEEE Trans. Reliab. **45**, 341–345 (1996)

23.20 I. H. Khamis: Optimum *m*-step, step-stress design with *k* stress variables, Comm. Stat. Comput. Simul. **26**, 1301–1313 (1997)

23.21 K. P. Yeo, L. C. Tang: Planning step-stress life-test with a target acceleration-factor, IEEE Trans. Reliab. **48**, 61–67 (1999)

23.22 S. J. Park, B. J. Yum: Optimal design of accelerated life tests under modified stress loading methods, J. Appl. Stat. **25**, 41–62 (1998)

23.23 M. H. Degroot, P. K. Goel: Bayesian estimation, optimal designs in partially accelerated life testing, Naval Res. Log. Q. **26**, 223–235 (1979)

23.24 L. C. Tang: Multiple steps step-stress accelerated test. In: *Handbook of Reliability Engineering*, ed. by H. Pham (Springer, London 2003) Chap. 24, pp. 441–455

23.25 J. I. Park, B. J. Yum: Optimal design of accelerated degradation tests for estimating mean lifetime at the use condition, Eng. Optim. **28**, 199–230 (1997)

23.26 H. F. Yu, C. H. Chiao: An optimal designed degradation experiment for reliability improvement, IEEE Trans. Reliab. **51**, 427–433 (2002)

23.27 M. Boulanger, L. A. Escobar: Experimental design for a class of accelerated degradation tests, Technometrics **36**, 260–272 (1994)

23.28 H. F. Yu, S. T. Tseng: Designing a degradation experiment, Naval Res. Log. **46**, 689–706 (1999)

23.29 S. J. Wu, C. T. Chang: Optimal design of degradation tests in presence of cost constraint, Reliab. Eng. Syst. Saf. **76**, 109–115 (2002)

23.30 G. B. Yang, K. Yang: Accelerated degradation tests with tightened critical values, IEEE Trans. Reliab. **51**, 463–468 (2002)

23.31 L. C. Tang, G. Yang, M. Xie: Planning step-stress accelerated degradation test with precision constraint, Proc. Ann. Reliab. Maintainab. Symp., 338–342 (2004)

23.32 S.J. Park, B.J. Yum, S. Balamurali: Optimal design of step-stress degradation tests in the case of destructive measurement, Qual. Technol. Quant. Man. **1**, 105–124 (2004)

23.33 H.F. Yu, S.T. Tseng: Designing a degradation experiment with a reciprocal Weibull degradation rate, Qual. Technol. Quant. Man. **1**, 47–63 (2004)

24. End-to-End (E2E) Testing and Evaluation of High-Assurance Systems

U. S. Department of Defense (DoD) end-to-end (E2E) testing and evaluation (T&E) technology for high-assurance systems has evolved from specification and analysis of thin threads, through system scenarios, to scenario-driven system engineering including reliability, security, and safety assurance, as well as dynamic verification and validation. Currently, E2E T&E technology is entering its fourth generation and being applied to the development and verification of systems in service-oriented architectures (SOA) and web services (WS). The technology includes a series of techniques, including automated generation of thin threads from system scenarios; automated dependency analysis; completeness and consistency analysis based on condition–event pairs in the system specification; automated test-case generation based on verification patterns; test-case generation based on the topological structure of Boolean expressions; automated code generation for system execution as well as for simulation, automated reliability assurance based on the system design structure, dynamic policy specification, analysis, enforcement and simulation; automated state-model generation; automated sequence-diagram generation; model checking on system specifications; and model checking based on test-case generation. E2E T&E technology has been successfully applied to several DoD command-and-control applications as well civilian projects.

The Department of Defense (DoD) end-to-end testing and evaluation (E2E T&E) project started in 1999 when the DoD was involved in the largest testing project ever, i.e, year 2000 (Y2K) testing. During Y2K testing, it was discovered that, even though DoD had many testing guidelines, most of them only addressed unit testing, and few were available for integration testing, but Y2K testing involved mainly integration testing, and thus needed E2E T&E guidelines.

Even though many techniques, such as inspection and program verification, are available for evaluating system reliability and quality, testing was and is the primary means for reliability and quality assurance. Furthermore, in practice, integration testing is often the most time-consuming and expensive part of testing. It is common to find software development projects with 50–70% of effort on testing, and 50–70% of the testing effort on integration testing. A review of the literature on integration testing shows that most integration testing techniques are either a methodology, such as incremental integration, top-down, and bottom-up integration [24.1], or are based on specific language or design structures of the program under test [24.2–5]. These techniques are useful, but are applicable to software written using the related techniques only. For example, an integration testing technique for an object-oriented (OO) program using Java may not be applicable to the testing of a legacy program using the common business-oriented language (COBOL). It may not be applicable to a C++ program because Java has no pointers but C++ does.

Due to these considerations, DoD initiated a project on E2E T&E in 1999 [24.6], intended to verify the interconnected subsystems as well as the integrated system. E2E T&E is different from module testing where the focus is on individual modules and is similar to, yet different from, integration testing where the focus is on the interactions among subsets of modules. Since 1999, E2E T&E has evolved from thin-thread specification and analysis, to scenario specification and analysis, and to scenario-driven system engineering (SDSE), and from SDSE to testing and verification of web services (WS) in a service-oriented architecture (SOA).

This paper is organized as following. Section 24.1 covers the history and evolution of E2E T&E and scenario specification. Section 24.2 presents an overview of the third and fourth generations of E2E T&E techniques. Section 24.3 elaborates static analyses, including model checking, completeness and consistency (C&C) analyses, and test-case generation. Section 24.4 presents automated test execution by distributed agents and how simulation of concurrent scenarios can be executed. Section 24.5 discusses policy specification and enforcement, which can be used to enforce safety and security policies, as well as dynamic verification and validation. Section 24.6 presents the reliability model for dynamic reliability assurance. Section 24.7 outlines the application of E2E T&E in SOA. Finally, Sect. 24.8 concludes this paper.

24.1 History and Evolution of E2E Testing and Evaluation

This section briefly describes the evolutionary development of the new generations of E2E T&E. Table 24.1 depicts the four generations of E2E T&E, their application periods, and the signature techniques in each generation.

24.1.1 Thin-Thread Specification and Analysis – the First Generation

The genesis of the DoD E2E T&E is thin-thread specification and analysis. This is based on the lesson learned from DoD Y2K testing. At that time, it was discovered that the DoD did not have an integration testing guideline that could be used for a variety of applications written in a variety of programming languages. Most existing integration techniques are either mainly of high-level concepts (such as those that used an incremental manner to perform integration testing) or are applicable to specific design structures or programming languages only, e.g., an integration testing techniques for object-oriented (OO) programs. Thus, there is an immediate need for an integration testing guideline that can be used by a majority of DoD organizations and services.

While no such DoD integration testing guidelines are available, it was discovered that most DoD organizations used the concept of thin threads to perform integration testing. A thin thread is essentially an execution sequence that connects multiple systems during system exercise and execution, and most organizations reported their Y2K testing effort in terms of the number of thin threads successfully executed and tested. In other words, thin threads were successfully used as the

Table 24.1 Evolution of E2E T&E techniques

Generations	Application period	Signature techniques
First	1999–2002	Thin thread specification and analysis techniques
Second	2001–2003	Scenario specification, analysis, and pattern verification techniques
Third	2003–present	Scenario specification and analysis, Scenario-driven system engineering, including reliability, security, and risk analysis; modeling and simulation
Fourth	2004–present	Scenario specification and analysis, and scenario-driven system engineering in service-oriented architecture with dynamic composition and recomposition

principal technique for Y2K integration testing. Integration testing based on thin threads has many advantages including:

- Thin threads are independent of any application;
- Thin threads are independent of any programming languages;
- Thin threads are also independent of any specific design structure;
- Thin threads can be used early during system development and late during system integration testing; and
- Thin threads can be easily understood by a vast number of DoD engineers. Thus, a DoD integration testing based on thin threads becomes a viable candidate for an integration guideline.

However close examination of DoD Y2K testing effort also revealed some important weakness of thin threads:

- Most thin threads were specified without a consistent format or using a localized format. In other words, different groups used different formats to specify thin threads;
- Most thin threads were developed manually and placed in an Excel file and thus were rather expensive to develop and maintain;
- The number of thin threads needed for successful integration testing was not known and thus some organizations used an extensive number of thin threads (such as thousands of thin threads) while some used only few (such as four to five) for a large application;
- The quality of thin threads was not easy to determine as they were developed manually and verified manually.

Thus, the first step of DoD E2E T&E focuses on the following issues:

1. The development of a consistent format for specifying thin threads. Because many thin threads share certain commonality with other thin threads, the DoD E2E guideline also suggests the organization of thin threads into a hierarchical thin-thread tree with related thin threads grouped together as a sub-tree in the thin-thread tree;
2. The development of a tool so that thin threads can be analyzed to ensure that these thin threads meet the minimum requirements;
3. The development of a guideline to determine the number of thin threads needed for an application; specifically, assurance-based testing (ABT) was developed to determine the number of thin threads needed for a certain system quality.

This first-generation DoD E2E T&E also assumes that each individual, participating system has been tested before they are subject to the DoD integration testing.

Several related techniques to thin threads have also been developed, including functional regression testing [24.7], automated dependency recognition and analysis, risk analysis, and test coverage based on specification of thin threads. Three versions of the DoD E2E tools have been developed and experimented with in the period 1999–2002. This experimentation showed that, once the thin-thread tree is specified, it is straightforward to develop test cases to run the integrated system.

24.1.2 Scenario Specification and Analysis – the Second Generation

During 2001–2002, experimentation of the DoD thin-thread tools revealed several serious shortcoming of thin threads:

- The number of thin threads needed is often too large to be manually developed even if an automated support tool is available; and
- Many thin threads differ only slightly from each other as they addressed the same similarity features and functionality of the application.

These shortcomings are due to the fact that each thin thread represents a specific execution sequence while a typical DoD application may have numerous execution

sequences. Thus, it is expensive and time-consuming to specify these thin threads even with automated tool support. To address these problems, it was discovered that it is possible to add control constructs such as *if-then-else* and *while* into thin threads. However, adding these constructs will change the meaning of thin threads because such a modified thread no longer represents one execution sequence, but multiple execution sequences. In other words, the modified thread is no longer a thin thread as defined by the DoD Y2K testing project, and the modified thread is called a scenario, for lack of an alternative, better names.

DoD E2E T&E is changed from the specification and analysis of thin threads to the specification and analysis of system scenarios; furthermore, techniques are developed so that thin threads will be automatically generated once system scenarios are specified. The fact that thin threads can be automatically generated from E2E scenarios makes them different from unified modeling language (UML) use cases. While UML use cases also describe system scenario from an external point of view, a use case does not need to be able to generate thin threads for testing, furthermore the E2E system scenarios can be used early in the development life cycle as well as late for integration testing and regression testing.

The original DoD E2E T&E techniques, such as automated dependency recognition and analysis, test coverage, risk analysis, functional regression testing, are modified so that they are applicable to the E2E system scenarios. Furthermore, several versions of tools were developed to support scenario specification and analysis.

Experiments with second-generation DoD E2E T&E were carried out on several projects, including a testing project for a high-availability communication processor. The requirements of a sample telecommunication processor were first translated into system scenarios, then the tool automatically generate a large number of thin threads from these scenarios, and finally an engineer developed test cases based on the thin threads generated. It was discovered that translating the original requirements into system scenarios is much easier than specifying the thin threads from the same requirements, and generating test cases from the thin threads generated is straightforward. The engineers involved in these experimentation also expressed the advantage of this approach over their current approach. It was much more difficult to develop test cases from the system requirements using their current approach, and the DoD E2E approach is much more structural and rigorous while saving them time and effort in developing test cases. In conclusion, second-generation DoD E2E T&E achieved its original goal of assisting test engineers to perform integration testing efficiently and effectively.

Several other new techniques were developed and discovered:

1. The system scenarios can be specified formally and subjected to a variety of formal analyses, not just the dependency analysis and risk analysis developed in the first generation of the E2E tool.
2. Systems often exhibit patterns in their behavior and these patterns can be rather useful for automated test-script generation.
3. As the developers of DoD E2E T&E always suspected, E2E techniques can also be useful in design and analysis rather than for integration testing only. This was confirmed in early 2003, and after hearing the briefing of the E2E T&E, a DoD organization started using the E2E T&E techniques for specifying and analyzing its command-and-control system.

Formalized scenario specification

The system scenario in DoD E2E T&E can be formalized in two ways: the first concerns the elements in the scenario, while the second concerns the process aspect of the scenario. The first aspect is formalized using the the actor, condition, data, action, timing, and event (ACDATE) model [24.8]. Using the ACDATE model, the specification of the software under development is described by its five model elements and their relationships.

- An **actor** is a model element that represents a system or its component with a clear boundary that interacts with other actors.
- A **condition** is a predicate on data used to determine the course of a process taken by actors. Conditions can be preconditions and post-conditions representing external and internal conditions and situations. Internal conditions represent the states of all system objects of interest, and external conditions can be network and database connections
- A **data** is an information carrier that represents the internal status of actors.
- An **action** is a model element that represents an operational process to change the internal status of an actor. Actions are performed when the preconditions are satisfied and events occur. Typically an action is
  a brief atomic computation such as
  – Assignment: sets the value of a variable,

  - Call: calls an operation on a target object,
  - Create: creates a new object,
  - Destroy: destroys an object,
  - Return: returns a value to a caller.
  - Send: generates an event, outgoing data,
  - Terminate: self-destruction of the owning object.
- The **timing** is an attribute of an actor, data, condition, event and action or behavioral model elements that describe their static or dynamic time information.
- An **event** is a model element that represents an observable occurrence with no time duration. Events can be internal and external occurrences that impact on, or are generated by, system objects such as incoming data (inputs), external action, and internal method call/message.

Once the specification of the software under development is represented by these five components and their relationships, the execution steps can be constructed using control constructs such as *if-then-else* and *while-do*, as shown in the following example. Figure 24.1 illustrates a simple scenario: "when both the driver and passenger door are locked, if remote controller is pressed for unlock, then the driver door will be opened", in the design of a car alarm system. In this scenario, five actors, one condition, one data, and one action are used.

Once the system scenario are specified, model checking, test-case generation, automated code generation, policy-enforcement-based dynamic testing, and simulation can be performed.

An important attribute is that scenarios can be specified in a hierarchical manner. The tester can first specify system scenarios at the highest level of abstraction. Once obtained, scenarios can be decomposed to show low-level details. This process can continue until scenarios are detailed enough for the T&E purpose. Furthermore, scenarios can be organized in a scenario tree where a group of related scenarios form a high-level scenario group [24.9, 10]. This feature is useful for testing an system of systems (SoS) because it often has subcomponents that interact with each other, and some of these components are legacy systems while others may be new systems that have just been introduced. Organizing system scenarios in a hierarchical manner facilitates test reuse and matches the hierarchical structure of the SoS.

Pattern Analysis

Even though a system may have hundreds of thousand scenarios, it may have only a few scenario patterns. For example, a commercial defibrillator has hundreds of thousand of scenarios, however, most (95%) of these scenarios can be classified into just ***eight*** scenario patterns [24.11]:

- Basic pattern (40%),
- Key-event-driven pattern (15%),
- Timed key-event pattern (5%),
- Key-event time-sliced pattern (7%),
- Command–response pattern (8%),
- Look-back pattern (6%),
- Mode-switch patten (8%), and
- Interleaving pattern (6%).

This provides an excellent opportunity for rapid verification because scenarios that belong to the same pattern can be verified using the same mechanism, except perhaps with different parameters such as timing and state information. This can save significant time and effort for implementation of test scripts.

```
usingACTOR:Alarm
usingACTOR:Horn
usingACTOR:DriverDoor
usingACTOR:PassengerDoor
usingACTOR:Trunk
```
Actors

Condition

```
if(CONDITION:DriverDoor,DriverDoorIsLocked ||CONDITION:PassengerDoor,PassengerDoorIsLocked)
then
{
    do ACTION:Alarm,TurnOnAlarm|
    do ACTION:Horn.MakeHornBeepOnce(DATA:Horn,HornStatus)
}
else
{
    do ACTION:Horn.MakeHornBeepThreeTimes(DATA:Horn,HornStatus)
}
```
Action

Data

Fig. 24.1 A sample scenario in the ACDATE language. Once the system scenarios are specified, model checking, test-case generation, automated code generation, policy-enforcement-based dynamic testing, and simulation can be performed

For example, suppose a system has 7000 scenarios, and if 15% of these scenarios belong to a specific pattern, these 1050 (7000×0.15) scenarios can be tested using the same verification software with individualized parameters. Thus, the productivity gain can be significant and industrial application of this approach showed that 25–90% effort reduction is possible [24.12].

Another significant advantage of this approach is the size reduction achieved using this approach. Industrial applications and experiments have indicated that the average length of code for test scripts reduced from 1380 lines of code (LOC) per scenario to 143 LOC per scenario using this approach, corresponding to a size reduction of 89.6%. If we assume that an expert test engineer can develop 1000 LOC of test script each week, the effort reduction achieved by using this approach is significant.

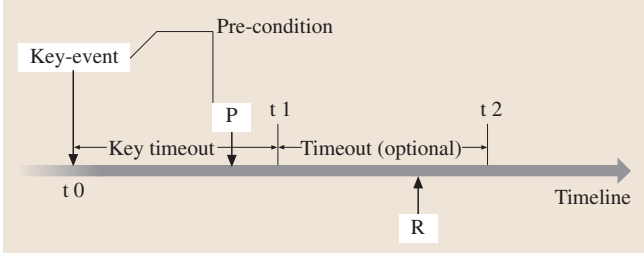


Fig. 24.2 Timed key-event-driven requirement pattern

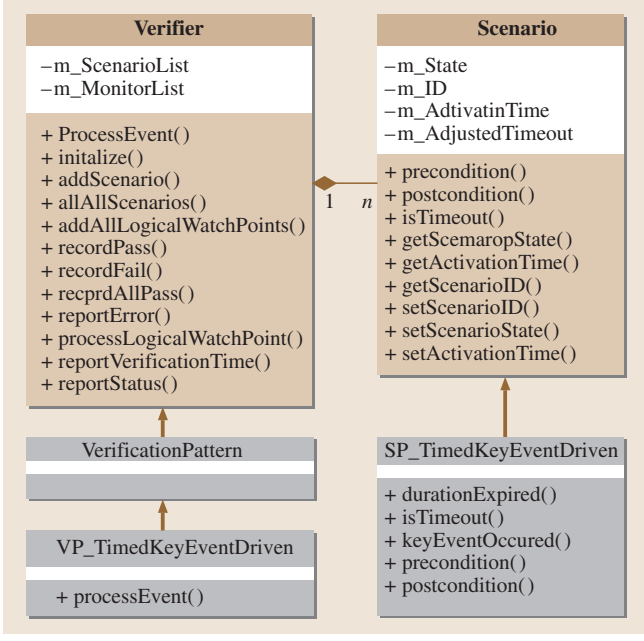


Fig. 24.3 Class diagram of the timed key-event-driven requirement pattern

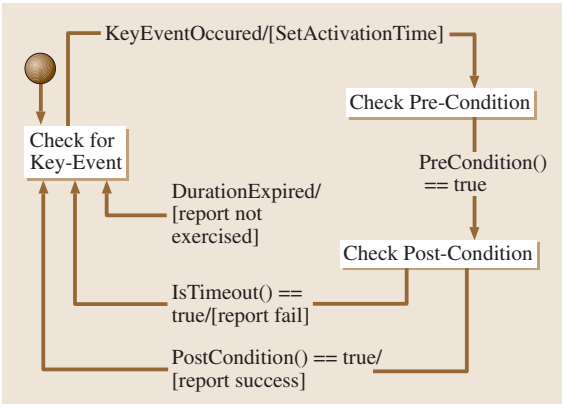


Fig. 24.4 Timed key-event-driven verification pattern

The following illustrate this concept for a timed key-event pattern.

Figure 24.2 shows a timed key-event-driven scenario pattern that includes two timing constraints for three events:

- Within the duration from t0 to t1 and after the key event, if event P occurs, then event R is expected to occur before t2.

As a typical example in an implantable defibrillator, when the device detects a heart problem, the capacity must be charged before it can apply a therapy to the patience, and this scenario shows three events (detection, capacity charged, and therapy applied), and the timing constraints between these three events.

Timed key-event verification patterns.
Name. Timed key-event-driven verification pattern
Description. The timed key-event-driven verification pattern is used to verify requirements that can be represented using the timed key-event-driven scenario pattern shown in Fig. 24.3. It provides an interface to decide if the duration has expired.
Verification state machine. Unlike the basic verification pattern, which starts checking the pre-condition right away, the verification process here checks the pre-condition within the *duration* after the key event occurs. The verifier can report "*not exercised*" if it failed to verify the pre-condition within this duration. Figure 24.4 shows an example of the timed key-event-driven verification pattern.

24.1.3 Scenario-Driven System Engineering – the Third Generation

Once system scenarios are formalized and experimented, DoD realized another need, i. e., can DoD E2E technology be useful for system engineering and system development? Traditional system engineering focuses on the following aspects:

- Reliability analysis;
- Safety analysis;
- Security analysis;
- Simulation, including distributed simulation and code generation;
- Verification and validation (V&V).

The DoD E2E already focused on V&V, and thus the quest is to extend the E2E technology to address the other aspects of system engineering. The rest of the paper will focus on

- Distributed simulation and code generation will be discussed in Sect. 24.4;
- Safety analysis: traditional safety analysis includes event analysis, event sequence and fault-tree analysis, while modern safety analysis includes static model checking and dynamic simulation analysis using executable policies. Model checking will be discussed in Sect. 24.1 and policy specification and enforcement will be discussed in Sect. 24.6;
- The DoD E2E security analysis is based on specification security policies and uses the simulation to evaluate the system vulnerability by verifying the security policies at runtime. These will be discussed in Sects. 24.4 and 24.6.
- Reliability analysis: it turns out that system scenarios are useful for both static and dynamical reliability analyses, and this will covered in Sect. 24.6.

24.1.4 E2E on Service-Oriented Architecture – the Fourth Generation

Currently, E2E T&E technology is being applied to the emerging SOA and web services (WS) platforms where more dynamic features are required, including dynamic composition, recomposition, configuration, reconfiguration, V&V, reliability assurance, ranking of WS, and methodologies that assess the WS. The fourth generation of E2E T&E has the same basic techniques but is implemented on a different software architecture. The basic techniques in the third and fourth generations of E2E T&E will be discussed in Sects. 24.3–24.7 and the SOA-specific techniques will be presented in Sect. 24.8.

24.2 Overview of the Third and Fourth Generations of the E2E T&E

This section outlines the major components of the basic techniques in the third and fourth generations of the E2E T&E. As shown in Fig. 24.5, the development process starts from the user requirements, which

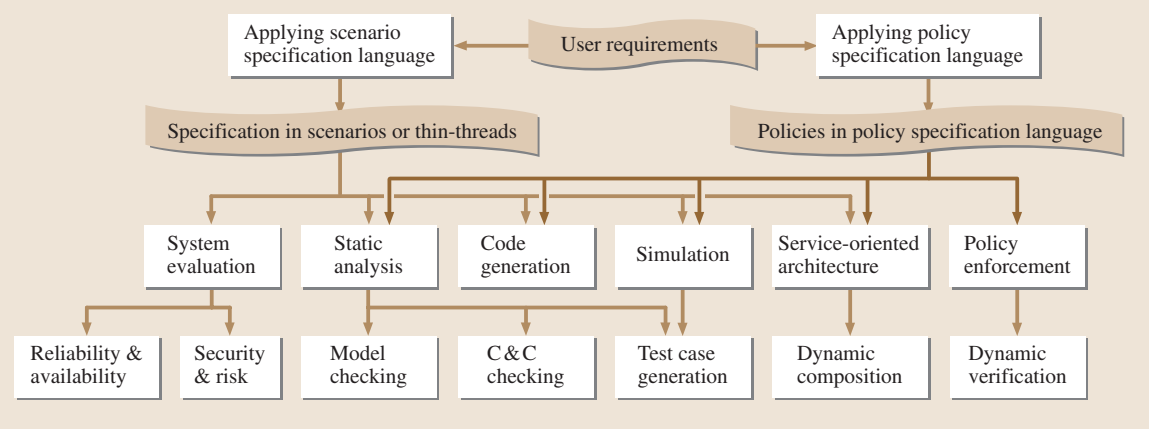


Fig. 24.5 The overall development and E2E T&E

is then formalized into the specification in the ACDATE scenario language, consisting of a sequence of actors, conditions, data, actions, timings, and events. Because scenarios are developed directly from the requirements, they are independent of any programming languages, design techniques, or development processes, such as the waterfall model, or agile development processes such as extreme programming [24.13].

Based on the scenario specification, static analysis, automated code generation, simulation, and system evaluation can be performed. E2E T&E also supports the service-oriented architecture (SOA) where software components are defined by standard interfaces that allow dynamic composition of new services based on existing services. On the other hand, the policies can be extracted from the user requirements and presented in a policy specification language. A policy-based system can be developed and policies are used to verify the behavior of the system dynamically. Static analysis, simulation, and code generation can be applied on policy specification.

Static analysis. Once the system is specified in scenarios, various analysis techniques [24.9, 14, 15] can be used to statically analyze the specification, for example, model checking and C&C. These analyses help the designer to make informed and intelligent decision in the requirement and design phases of project development [24.7, 13, 15]. In other words, the E2E T&E tool can be used early in the life cycle, during, and throughout the rest of the life cycle, including during operation and maintenance. Several methods can be applied to perform model checking, for example, Berkeley lazy abstraction software verification tool (BLAST) [24.16] developed at the University of California at Berkeley and C&C analysis [24.11]. In the process of model checking, both positive and negative test cases can be generated [24.17]. The positive test cases are used to test if the system generates correct output for valid inputs, while the negative test cases are used to test if the system does not generate an undesired output. An undesired output is one that could cause an undetected error or an unacceptable consequence.

The E2E process also supports rapid test-case generation by classifying system scenarios into patterns, where each pattern has a corresponding test case that can be parameterized to test all the system scenarios belong to the pattern. This approach promotes test-case reusability and reduces the cost of test-script generation significantly [24.18]. This approach has been used successfully to test commercial real-time safety-critical embedded medical devices such as pacemakers and defibrillators.

Code generation. Once the model is verified, the automated code-generation tool can be applied to generate the executable directly from the scenario specification. Code generation can also be performed in the simulation framework.

Simulation. Simulation is a practical way to prove the design idea and assess the performance of complex systems dynamically [24.19]. The traditional simulation is done in a specify-and-code or model-and-code manner, which means that, in the simulation process, engineers first construct the target system model and then develop the simulation code to run the simulation. This approach is expensive and inflexible. The automated code generation in E2E T&E environment can perform model-and-run or specify-and-run simulation. In other words, once the scenario model is constructed, no additional simulation coding effort is needed to run the simulation. Even the real target system's code can be automatically generated from the system model with little or no human involvement, because the same automated code-generation tool is applied to generate the real system code and simulation code. In E2E simulation, once scenarios are available, the system is executed in a simulation environment by tracing the conditions and actions in the scenarios. Simulation can be used together with other analyses to prove the ideas and features of the system design. For example, simulation can be used together with timing analysis to determine if the system satisfies the timing requirements. Furthermore, multiple scenarios can be simulated at the same time to determine the interaction of these scenarios. The E2E tool supports distributed test execution by providing the architecture with a test master and test agents. The test master is responsible for managing test scenarios and test scripts, and sending test commands to test agents for remote execution. Test agents are responsible for sending test commands to the system under test for test execution, collecting and data analysis, and for reporting test results to the test master.

Policy enforcement and dynamic verification. A policy-based system allows the requirements and the specification to be modified dynamically. The typical application of a policy-based system is in dynamic safety and security enforcement where the safety and security of a system can change from time to time. A policy-based system is also useful when the system is dealing

with a dynamically changing environment or the functional requirements can change from time to time. For example, the initial system has been programmed for an application temperature range of 0–100 degrees. If the range is later extended to 0–150 degrees, a policy-based implementation does not need to modify the program code and regenerate the executable code. Only the policy data need be updated and reloaded. Policy enforcement can be applied as a dynamic V&V method. In fact, many functional requirements of a system can be extracted as policy requirements. For example, array index range checking, probability value range checking, and execution-order checking can be written as policies. As a result, policy enforcement can dynamically check the validity of computing. Policy enforcement is particularly useful in detecting bugs that are difficult to catch during unit testing and those with complicated interactions due to concurrent threads and processes in simulation.

System evaluation. E2E T&E supports both static and dynamic analyses of reliability, availability, security, risk, timing, usage, dependency, and the effectiveness of test cases. The evaluation results can be applied immediately to guide subsequent testing. For example, according to the number of faults each test case detects, the effectiveness of the test cases can be ranked dynamically. In subsequent testing, the more effective test cases will be applied first in testing. The reliability evaluation results can be applied in selecting components that need to be replaced dynamically.

Service-oriented architecture. The service-oriented architecture (SOA) considers a software system consisting of a collection of loosely coupled services. These services can make use of each other services to achieve their own desired goals and end results. Simple services can cooperate in this way to form a complex service. Technically, a service is the interface between the producer and the consumer. From the producer's point of view, a service is a function that is well defined, self-contained, and does not depend on the context or state of other functions. In this sense a service is often referred to as a service agent. The services can be newly developed applications or just wrapped around existing legacy software to give them new interfaces. From the consumer's point of view, a service is a unit of work done by a service provider to achieve desired end results for a consumer. The next generation of E2E T&E will deal with SOA and composition and recomposition, dynamic configuration and reconfiguration of software systems. Initial investigations have been performed [24.14, 17, 20, 21].

24.3 Static Analyses

To ensure the correctness of the specification, static analysis will be performed, including model checking and C&C analysis. Test cases can be generated in the process of static analysis.

24.3.1 Model Checking

Model checking has been proposed recently to facilitate software testing following the idea that model checking verifies the model while testing validates the correspondence between the model and the system. One of the most promising approaches was proposed at the University of California at Berkeley using BLAST [24.16]. The BLAST model checker is capable of checking safety temporal properties, predicate-bound properties (in a form that asserts that, at a location l, a predicate p is true or false), and identify dead code. BLAST abstracts each execution path as a set of predicates (or conditions) and then these predicates are used to generate test cases to verify programs. This approach is attractive because it deals with code directly rather than the state model used in traditional model checking [24.22]. Thus, the BLAST approach is better suited for software verification than traditional model checking. However, BLAST does not handle currency and its test-case generation is targeted mainly on the positive aspects of testing. Negative aspects such as near misses are not handled. In our E2E T&E framework, many scenarios may be active at the same time, and it is necessary to verify that concurrent execution of these scenarios will not cause the system to deviate from its intended behavior. We extend the BLAST approach to suit the scenario specification in three ways: (1) instead of using the source code to drive model checking, we use our scenario modeling language for model checking. The control-flow automata used by BLAST resembles the workflow model derived by the control constructs in the scenario language; (2) we rely on the conditional or unconditional output, effect, and precondition in each thin thread to construct their essential inner control logic; and (3) we enhance

BLAST to handle concurrent executions of processes in the ACDATE language [24.17].

24.3.2 Completeness and Consistency Analysis

Software requirements are often incomplete, inconsistent, and ambiguous. The specification based on the requirements may have inherited the faults. Faults introduced in this stage of development have been shown to be difficult and more expensive to correct than faults introduced later in the life cycle. C&C analysis on specification aims to eliminate requirement- and specification-related faults. As shown in Fig. 24.6, the process starts from converting user requirements into a ACDATE scenario specification; extracting condition and event (CE) combinations from the specification; performing a completeness analysis to identify all the missing CE combinations; performing consistency analysis to check if the CE combinations are consistent with each other; and identifying the set of scenarios that need to be modified to make the system reliable and robust. More specifically, these steps are explained as follows.

1. Derive system scenarios from the system requirements: formalize system scenarios using the ACDATE model, which includes elements (actor, condition, data, action, timing and event) and the relations among them.
2. Parse each scenario and extract the combinations of conditions and events: from the ACDATE model, the CE combinations are automatically extracted. The CE combinations are partitioned into independent components where each component does not interact or related to the others. Two scenarios are independent of each other if there is no way for them to interact or influence each other. For example, two scenarios that share a common condition are considered related, and the related relationship is transitive. By exhaustively examining the transitive relationships, one can determine if two scenarios are independent of each other.
3. Perform C&C analysis on CE combinations: once the CE combinations are obtained in step 2, consistency analysis on CE combinations is performed and completeness analysis on CE combinations is then performed to identify those missing CE combinations.
4. Construct patching scenarios to eliminate those missing CE combinations. Using an Karnaugh-map analysis, we can aggregate a large number of missing CE combinations into a smaller set of equivalent CE combinations. From these CE combinations, we can develop patching scenarios to cover the missing condition.
5. Classify each patching scenario into one of the three categories: (1) incorporate it as a functional scenario; (2) treat it as an exception with an exception handling; or (3) consider it as a do not care item, based on the nature of the application. In the first case, the covering scenario is indeed an intended behavior but missed in the specification, in the second case the covering scenario is not intended and should be masked out in the specification.
6. Amend the scenario specification using the C&C analysis results: use the results in step 5 to patch the scenario specification automatically.
7. Inform the user about the amendment of the specification and seek amendment of the requirements from the user.

The C&C analysis process is an iterative and incremental process. After each amendment, the C&C process should be repeated to ensure that the amendment does not introduce new consistency. Tools have

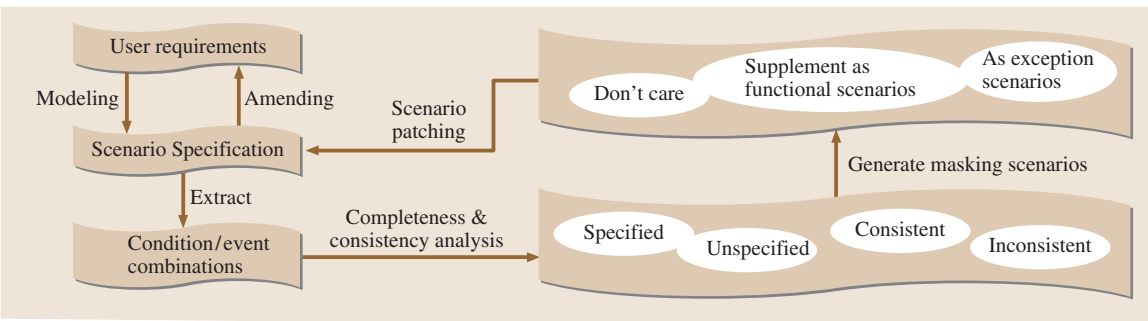


Fig. 24.6 The process of C&C analysis

been developed to perform steps 2–6 automatically. Experiments with the tools in several large, industrial applications have been carried out and the results indicate that the process described above is feasible and scalable to large applications.

24.3.3 Test-Case Generation

Test-case generation techniques can be greatly enhanced by comprehensive formal C&C analysis followed by test-case generation based on Boolean expressions [24.11]. An important distinction of this approach is that test-case generation is based on the quantitative Hamming distance. All previous approaches, including modified condition/decision coverage (MC/DC) and MUMCUT [24.23], were based on user experience and intuition. Exploring the topological hypercube structure of Boolean expressions can easily reveal the faults not discoverable by previous approaches. Furthermore, these two mechanisms can be completely automated, thus saving significant effort and time. After the Boolean-expression generation, the Swiss-cheese test-case-generation tool can be applied to obtain both positive and negative test cases.

The Swiss-cheese (SC) approach is an efficient iterative algorithm developed based on C&C analysis [24.17]. It can identify most error-sensitive positive test cases and most critical negative test cases. Given the Boolean expressions that represent the system specification, the algorithm first maps the Boolean expressions into a multidimensional Karnaugh map called a polyhedron. The algorithm then iteratively identifies all boundary cells of the polyhedron and selects the most error-sensitive test cases among all the boundary cells. The more neighboring negative test cases (degree of vertex – DoV) a boundary cell has, the more error-sensitive it is. The last step is post-checking, which tries to identify critical negative test cases within the polyhedron. For each negative test case, the term Hamming distance (HD) is used to define the minimum different Boolean digits between it and any boundary cells. The HD of all boundary cells is 0. The smaller the HD is, the more critical a negative test case is. It is shown in this paper that negative test cases can detect more failures. The SC approach uses the most critical negative test cases first to test a program, and then randomly chooses the remaining test cases.

24.4 E2E Distributed Simulation Framework

Traditional simulation methodologies adopt a model-code-run approach, such as that used in the Institute of Electrical and Electronics Engineers (IEEE) modeling and simulation (M&S) high-level architecture (HLA) [24.24] and other popular simulation frameworks, which means that the engineers must create a model of the target system, develop the simulation code based on the model, and then run the simulation code, as discussed in GALATEA [24.25]. The E2E scenario-based modeling and simulation framework pro-

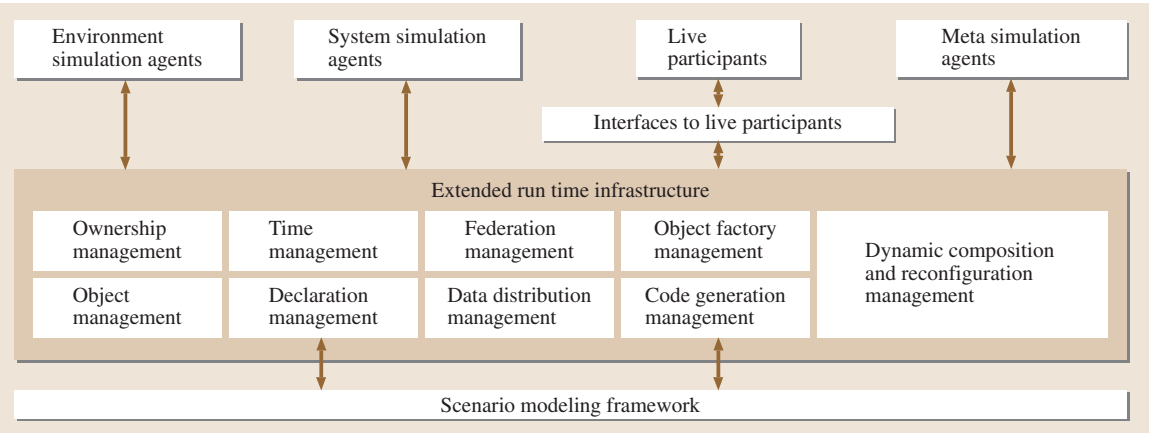


Fig. 24.7 Simulation framework architecture

vides a model-and-run paradigm for simulation. In other words, once the model is available, the model is directly executed for simulation without manual simulation-code development. The simulation code is automatically generated from the system scenario specification. Before the code is generated, the model can be evaluated using existing E2E analysis techniques such as C&C analysis to ensure that the model is correct. Furthermore, the simulation run can be dynamically verified using a formal policy specification.

24.4.1 Simulation Framework Architecture

Figure 24.7 shows the E2E T&E simulation framework running on an SOA, based on software agents. According to the definition in [24.26, 27], a software agent is "an autonomous computer program that operates on behalf of something or someone else" and can "be viewed as a self-contained, concurrently executing thread of control that encapsulates some state and communicates with its environment and possibly other agents via some sort of message passing". The agents here serve as an entity that is capable of carrying out the simulation task and performing a variety of analyses. Both the agents and the simulation framework are designed according to an object-oriented layout to support distribution (of objects/agents), modularity, scalability, and interactivity [24.25], as demanded by the IEEE HLA specification [24.24].

The E2E simulation framework integrates the concepts and tools that support modeling and simulating systems under the distributed, interactive, continuous, discrete, and synthetic focuses. The simulation framework consists of:

- The ACDATE scenario language and the framework that allows the construction of the system models.
- The on-demand automated dynamic code generator that supports rapid and automated simulation/real system-code generation such that simulation can be carried out once the system model is ready. No additional programming effort is needed. The *execution* here means that the real system components' execution is involved in the system simulation, i. e. end-to-end simulation including the end hardware-in-the-loop and man-in-the-loop.
- Simulation agents that carry out the simulation tasks and form a simulation federation (in the HLA sense) serve as the simulator for the whole system. These agents can be geographically distributed on computers that are interconnected via a local-area and/or wide-area network [24.19].
- An extended runtime infrastructure to support the agents' work. As required in [24.28], the simulation here is separated from the target system model, which makes the simulation framework flexible and generic.

As discussed in [24.24], traditional simulation techniques should be extended to support interactive simulation of a number of programs executing in heterogeneous and distributed computers that interact with each other through communication networks and are managed by a distributed operating system. The IEEE has provided the HLA framework to allow the development of a standard simulation framework with many different simulation components, which is used as a reference for the design of our framework. Figure 24.7 shows the architecture of our simulation framework. As can be seen, the scenario modeling framework provides the scenario specifications of the target systems. The extended runtime infrastructure separates the simulator (which consists of the agents and/or the live participants) from the target system model and provides the necessary runtime support for the simulator.

24.4.2 Simulation Agents' Architecture

The E2E T&E simulation framework is object-oriented, agent-based, discrete-event-driven, distributed, and real-time. In object-oriented terms, E2E simulation is based on the integrated ACDATE scenario model, which is based on SoS/SOA and the object-oriented modeling methodology. Each component in the system is modeled as a specific object–actor that has interfaces (actions), behaviors (scenarios) and constraints (policies). The simulation is carried out by a set of simulation agents. The agents are the most important elements in our simulation framework.

An agent can simulate either a single actor or multiple actors. Two agents may or may not reside in the same computation site. Agents can talk with each other via standard communication protocols. The behavior of a simulation agent is determined by the SoS/SOA scenario model of the actors simulated on this simulation agent. Based on the system's scenario model, it is clear how an actor will behave to some outside stimulus either from the environment or from some other agents under given conditions. The outside stimuli are modeled as discrete events that can be received and processed by an actor. Once an event arrives at an actor, the ac-

tor will put it into its waiting queue. How the events in the waiting queue are processed depends on the actor's scenario model, i. e. if the system is modeled as a multi-tasking actor, any incoming event can be processed as long as there is enough resource. If the system is modeled as a single-tasking actor, an incoming event can be processed only if no other task is scheduled to use the processor, and so on. Due to some uses of the framework for decision-making, one simulation run should finish before a given deadline, if required.

The simulation can be formally described as follows. Simulation of an actor A_i starts from the point when an event $E_{i,k}$ arrives at A_i. At the point, A_i will pick up scenario $\text{Scnr}_{i,w}$, which describes the behavior of A_i in response to $E_{i,k}$ and sends $\text{Scnr}_{i,w}$ to a scenario simulation/execution engine, as shown in Fig. 24.8. The scenario simulation engine will then interpret $\text{Scnr}_{i,w}$ and perform the following:

- Check current system condition, which includes the A_i own conditions $\{C_{i,i1}, C_{i,i2}, \ldots, C_{i,im}\} \in \{C_{i,0}, C_{i,1}, \ldots, C_{i,Mi}\}$ and/or other actors' conditions as guard conditions.
- Based on the system condition and chosen scenario $\text{Scnr}_{i,w}$, the simulation engine will choose an execution path which includes a series of actions $\{\text{Act}_{i,v1}, \text{Act}_{i,v2}, \ldots, \text{Act}_{i,vn}\}$.
- The scenario simulation engine will carry out the chosen actions, whose semantics are also specified using scenarios. Thus, whether an action can be successfully performed also depends on the system conditions at that point. An action may change the owner actor's status by changing the values of the data owned by the actor; or emit a new event either to other actors or to its owner actor.
- Agents' states will be changed accordingly as the actions are performed, which is reflected in the data-changing function: Act: $D \rightarrow D_0$, where D is the set of data values before the action Act is performed, where D_0 is the set of data values after the action Act is performed.

Events are the only channel through which different actors can communicate with each other. An event can carry parameters to provide more information for the receiver to make decision on how to respond to the incoming event. Simulation agents used here contain versatile communication capability, which is implemented by the communication component of each agent, and thus an agent can be exposed to the outside world as a traditional transmission control protocol/internet protocol (TCP/IP) service, a dedicated network component,

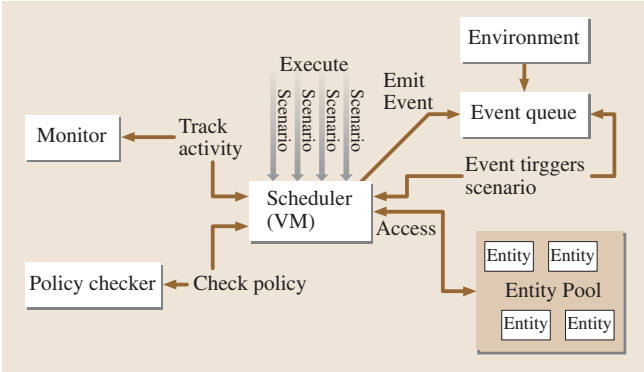


Fig. 24.8 Simulation engine inside a distributed simulation agent

or more generally a web service. In the latter case, as each simulation agent is exposed as a web service, it is easy for the simulation users to perform the simulation tasks on the internet.

There are three major types of simulation agents: environment simulation agents, system simulation agents and meta-agents. By separating the environment simulation agents and system simulation agents, it is easy to study the target system's behaviors under different environments without touching the target system model and simulation, which only requires a change to different environment simulation agents in the simulation. Meta-agents are agents that monitor and coordinate the whole simulation. With the help of these meta-agents, engineers can easily know what is going on in the distributed simulation from a global point of view. These meta-agents can also help perform dynamic analyses that involve more than one participating simulation agents such as the generation of overall system behavior.

24.4.3 Simulation Framework and Its Runtime Infrastructure (RTI) Services

The simulation extended runtime infrastructure is an extension and enhancement of high level architecture/runtime infrastructure (HLA/RTI) [24.24], which serves as a design reference for our simulation framework's runtime infrastructure. The major improvements are the automated ACDATE/scenario code generation and deployment management, event management on SOA, and automated simulation runtime reconfiguration and recomposition. With the help of these services, our simulation framework is capable of providing on-demand simulation, which means that the simulation code can be dynamically obtained and used for simulation from the dynamic code generator whenever it is

demanded by the users; as well as dynamic simulation reconfiguration.

In contrast to traditional HLA/RTI services, which are exposed as traditional remote procedure call (RPC) methods using the user datagram protocol (UDP)/TCP, the services provided by our simulation framework can be exposed as either RPC-like service using binary communication data via TCP, or WS using simple object access protocol (SOAP) messages to carry communication data via the hypertext transfer protocol (HTTP).

Managing events in SOA

The simulation framework is developed on top of an SOA, and thus it can reuse resources on SOA, a lot of benefits can be obtained. One of these is that a simulation agent does not need to know the existence of other simulation agents. Simulation framework RTI will provide an event-space service (ESS) to facilitate communication among simulation agents, as shown in Fig. 24.9. The services provided by ESS include:

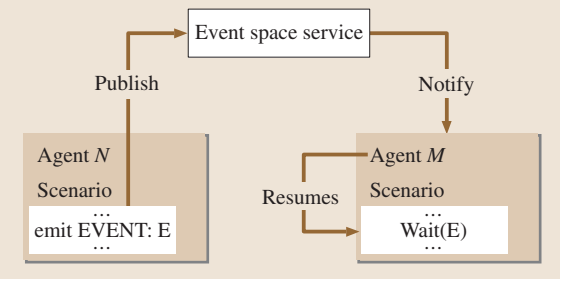


Fig. 24.9 Event publishing and notification example

- Event registration
  - Event publishing registration: before sending out any event, agents must register what events they will send out with the ESS.
  - Event subscription registration: an agent must subscribe the interested events before it can actually know that the event happens.
- Event publishing: agents can emit events using ESS
- Event notification: ESS can notify the occurrence of events to those agents that have subscribed the events.

Automated Simulation Code Generation and Deployment

The simulation code is generated based on the scenario specification, which includes the ACDATE definition and scenario description, as shown in Fig. 24.10. Each ACDATE element will be translated into an object with the attributes defined in the specification. Instrumentation code will be inserted into the objects to interface with the monitor and policy checker. Each scenario will be translated into a procedure that is basically a sequence of operations on the ACDATE objects or emitting events. Similarly, instrumentation code will be inserted into the procedure so that the procedure can interface with the scheduler to schedule concurrent execution and the event queue for emitting new events.

Table 24.2 shows a sample simulation code automatically generated with instrumentation code that interfaces with the scheduler, event queue, monitor, and policy checker.

Simulation framework provides two base components for scenario code generation: BaseACDATE

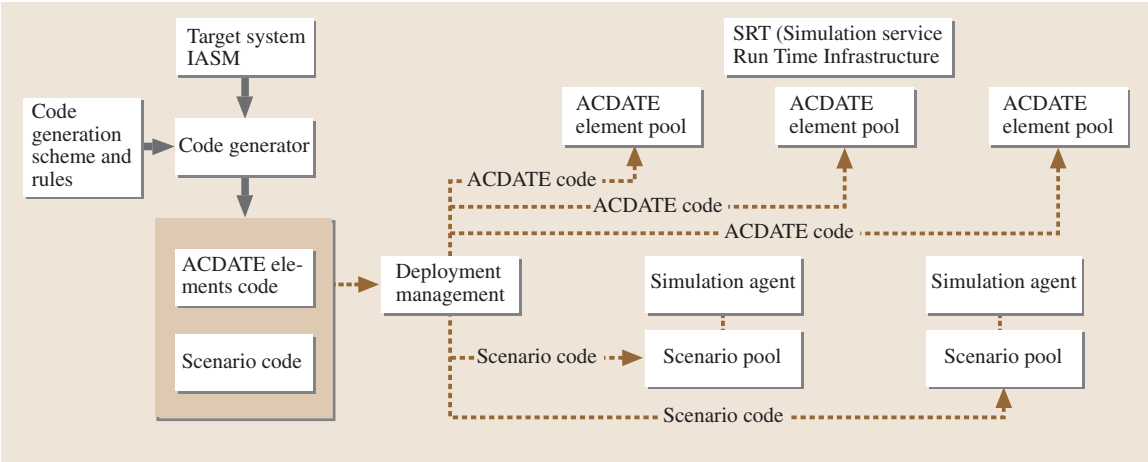


Fig. 24.10 Automated simulation-code generation and deployment

Table 24.2 Automatically generated code example

```
public override void ScnrFunc()// a scenario
 . . . . . .
{
Condition_4 condition4 = new Condition_4();// obtain ACDATE elements
Action_5 action5 = new Action_5();
Data_2 data2 = new Data_2();
e = newSimRunTimeLogArgs(SimRunTimeLogArgs.LogTypes.ScnrPreStatement,2,
1, 0, "Before Step 1 in scenario 6");
this.OnPreScnrStatementEventHandler(this, e);
data2.SetValue(1); // data2's value is now changed to integer 1
e = newSimRunTimeLogArgs(SimRunTimeLogArgs.LogTypes.DataWrite,
2, 1, 0, "After Step 1 in scenario 6");
// interface to instrumentations such as policy checker embedded here
this.OnPostScnrStatementEventHandler(this, e);
System.Windows.Forms.MessageBox.Show("Agent 1 - Step 1 done");
e = newSimRunTimeLogArgs(SimRunTimeLogArgs.LogTypes.SchedulingFlag,
System.Threading.Thread.CurrentThread.GetHashCode(), 1, 0, "Calling
Scheduling");
this.OnSchedulingEventHandler(this, e);
System.Threading.Thread.CurrentThread.Suspend();// interface to simula-
tion scheduler
 . . . . . .
}
```

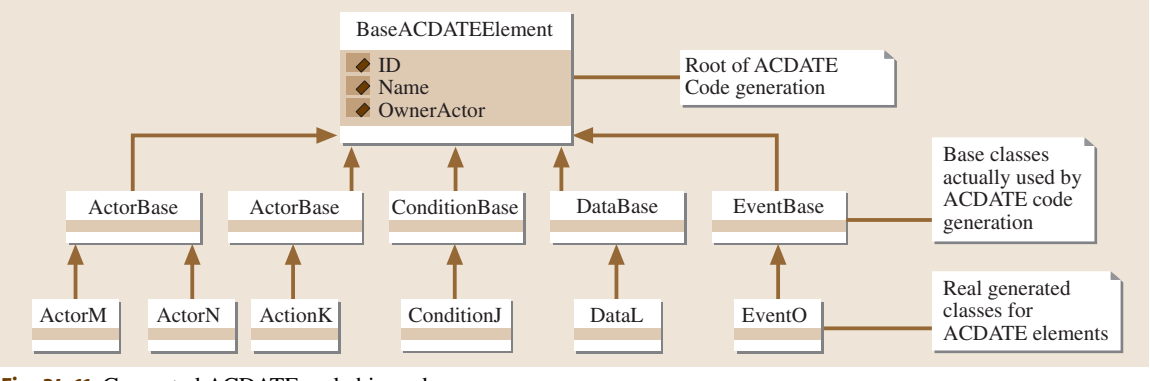


Fig. 24.11 Generated ACDATE code hierarchy

(Fig. 24.11) and BaseScenario (Fig. 24.12). The BaseACDATE component contains all the base class definitions for the ACDATE elements in the scenario. The BaseScenario component provides the base class for the scenario specification in the scenario model while referencing the used ACDATE elements' information in the BaseACDATE component.

For different system simulations, different simulation code will be automatically generated based on the target system's scenario model. The generated code is divided into two major categories of components: ConcreteACDATE and ConcreteScenarios. ConcreteACDATE here does not mean a single component but a collection of components holding

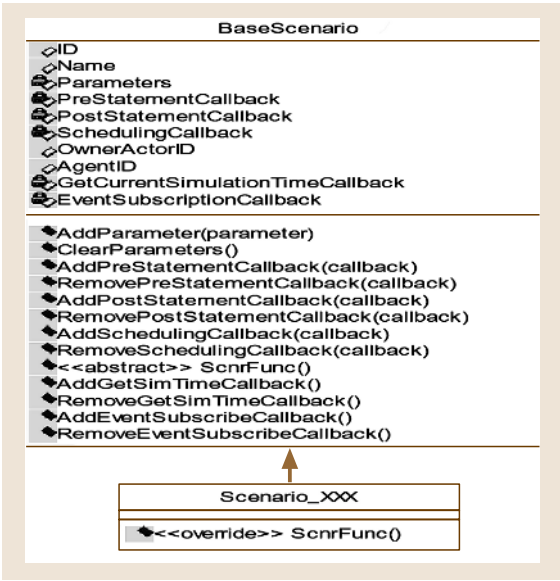


Fig. 24.12 Generated scenario code hierarchy

the generated code for the concrete ACDATE elements in the target system's scenario model. Similarly, ConcreteScenarios is a collection of components holding the concrete scenario code of the target system's scenario model, with only references to the related ConcreteACDATE components. Each generated component has its own deployment configuration document, based on which the deployment management will deploy the simulation code properly. More details of automated simulation code generation will be discussed in later sections.

At simulation runtime, with different simulation code loaded, the simulation agents can simulate different target systems while keeping the simulation agents themselves untouched.

In a distributed simulation system, simulation deployment is an important issue. Traditionally simulation deployment is done manually, which is time-consuming and error-prone, especially for large simulation systems. Simulation deployment can be formally specified and then be automated. As has been discussed, with different target system models (code) loaded, the simulation agents can perform different simulation tasks. Machine understandable simulation deployment specification [in extensible markup language (XML)] will be provided. Service RTI will provide a dynamic model (code) load/unload service based on the simulation deployment specifications.

Automated Simulation Reconfiguration and Recomposition

For a large-scale distributed simulation, a single failure may corrupt the whole simulation if the failure point is critical (a single point of failure). It is important to organize the simulation services (agents) distributing across the internet to form a functional simulator, which can make use of the unutilized computation power. With the policy and dynamic reconfiguration service (DRS), the simulation framework should be able to see the change of system behavior when a new policy becomes effective during simulation without shutting down the system.

Dynamic simulation composition and reconfiguration management involves the following issues:

- Automated simulation-agent deployment and discovery
- Automated simulation-agent status monitoring and failure detection
- Automated dynamic simulation code generation
- Automated dynamic simulation configuration generation
- Automated dynamic simulation deployment and redeployment.

A simulation agent knows nothing about the target system until the corresponding simulation code is loaded into the agent. With different simulation code loaded, the simulation is capable of simulating different target systems. The simulation agent can also unload the previously loaded simulation code component and reload a new set of simulation code components to simulate another target system. In this sense, the real components of a functional simulation are the dynamically and automatically generated simulation distributed across the network.

The first problem one may face is how the dynamically generated simulation code components can know where the counterparts and the runtime infrastructure services are. This can be solved with using a dynamically and automatically generated configuration of the simulation. With any given simulation topology, users can specify where and how a simulation should be deployed. A configuration document will then be automatically generated for each dynamically and automatically generated simulation code component based on the users' deployment requirements, such that each simulation code component can know where to obtain the required resources and services, as well as how to communicate with the runtime infrastructure services and its counterparts. This has been introduced in previous sections.

The automatically generated simulation code components along with their configuration documents are deployed based on the deployment configuration documents to set up the simulation agents properly with the help of the automated deployment management services, as shown in Fig. 24.13.

During the simulation, the status-monitoring services continuously monitor the status of the simulation agents involved. Once the failure of a simulation agent is detected, the runtime infrastructure will try to discover an available simulation agent and perform the code and configuration generation again for the alternative agent. The new simulation agent is then loaded with the simulation code and configuration and the simulation is resumed. Using automated simulation composition and deployment, users can also easily change the deployment of the simulation anytime. However, in these cases all the unfinished work on the crashed simulation agents will all be lost, if it has not been saved.

There is another scenario in which dynamic simulation reconfiguration can be used: on-the-fly model changing and continuous simulation. Users can change the target system model during the simulation without restarting the simulation to reflect the effect of model modification.

Once the target system model has been changed, based on the original users' deployment requirements and the status of the simulation agents, the automated simulation deployment service can determine which agents are affected by the model modification. The simulation code components and configuration documents are then regenerated based on the modified model. Once the affected simulation agents enter a safe state where unloading and reloading simulation code components will not affect other running simulation agents, the automated simulation deployment service will unload and reload the simulation agents with the modified simulation code components. Before the simulation components are unloaded, the status of the simulation agents is saved. The saved information is used to restore the original agents after the new simulation code components are reloaded. In this way, the users' simulation will not be interrupted by the model modification.

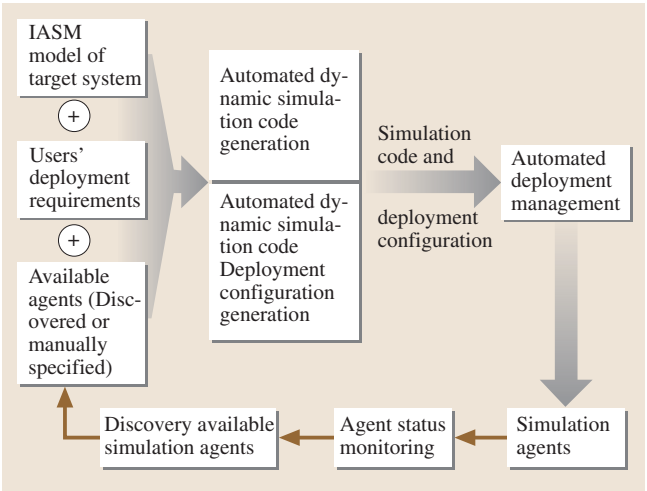


Fig. 24.13 Automated simulation reconfiguration and recomposition

24.5 Policy-Based System Development

Policies have been increasingly used in computing systems for specifying constraints on system status and system behaviors. A *policy* is a statement of the intent of the policy maker or the administrator of a computing system, specifying how he or she wants the system to be used [24.29]. Usually, policies are hard-coded into the system implementation, For instance, if a policy states that "passwords must be at least eight characters long", there must exist a snippet of code in the system implementation that checks the length of passwords. Hard-coded policies can cause major problems for the system such as:

- It is difficult and expensive to update.
  Whenever a policy needs to be changed (e.g. the system administrator wants to reduce the minimum length of valid passwords from eight characters long to five characters long), the whole system has to be shut down, and the code has to be modified, recompiled, and redeployed. The process is lengthy and significantly increases an organization's operating expenses. Shutting down a mission-critical system, in most cases, is prohibitive and may cause disastrous consequences to the mission.
- It is difficult to manage.
  Hard-coding policies do not separate policy specification from system implementation. Policies are spread throughout the system implementation. If a policy maker wants to know how many policies

there are in the system, or what are the policies that are defined for the role of supporting arms coordinator, there is no easy way to find the answers.

In the past decade, a number of *policy specification languages (PSLs)* have been proposed [24.29–33]. PSLs provide a simple and easy-to-use syntax to specify policies separately from the system implementation. A user interface can provide a means for policy makers to specify and manage policies. A policy engine can interpret and enforce policies at runtime rather than at compilation time, which allows policies to be dynamically added, removed, updated, and enabled or disabled.

24.5.1 Overview of E2E Policy Specification and Enforcement

Figure 24.14 shows the simulation environment that we developed for highly developed systems. When developing such systems, V&V needs to be performed at each step of the development. First, the system requirement is translated into the formal specification. The specification is verified by a C&C check. After several iterations of verification, the final specification is obtained. Test cases are generated from the specification. An automated code-generation tool generates the executable for simulation. This process has been reported in [24.17].

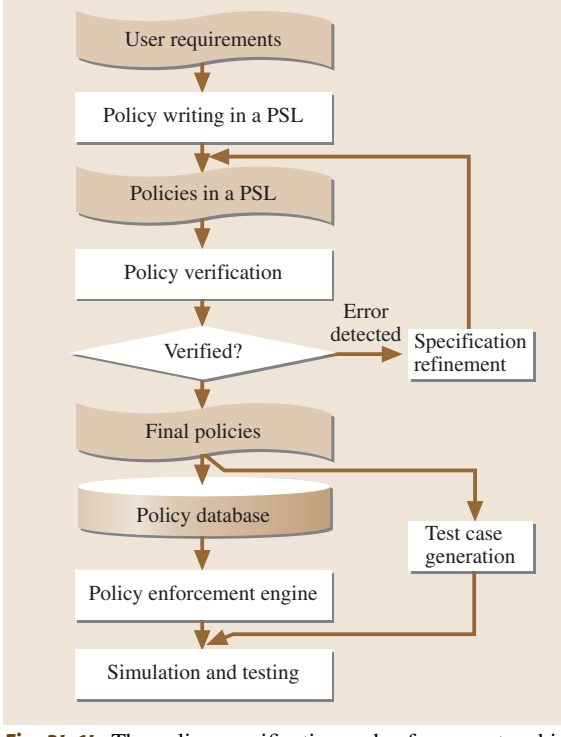


Fig. 24.14 The policy specification and enforcement architecture

Independently of the development processes, the policies are extracted from the requirements and then written in a policy specification language (PSL). Similarly to the specification, the policies are verified by a C&C check to detect any incomplete and inconsistent policies. After the policies pass the verification, they will be stored in a policy database. Test cases that dynamically check C&C can be generated from the final policies. During the course of simulation execution, a policy-enforcement engine dynamically loads the policies from the policy database, interprets them, and enforces them at runtime. Since the policy engine dynamically interprets and enforces policies, policies can be easily changed (added, removed, updated, and enabled/disabled) on-the-fly at any time.

This paper not only makes use of the flexibility of policies, but also applies policies as a dynamic V&V method. In fact, many functional requirements of a system can be extracted as policy requirements. For example, array index range check, probability value range check, and execution orders can be written as policies. As a result, the policy enforcement can dynamically check the validity of computing. Policy enforcement is particularly useful in detecting those bugs that are difficult to catch during unit testing and those with complicated interactions due to concurrent threads and processes in simulation. Note that traditional testing suffers from the need to set up the environment to a given state and run the program to see if the program behaves as intended, which is time-consuming and difficult. Policy is a good way of ensuring the simulation program is correct because an engineer can specify any kinds of policies that need to be enforced to see if the simulation program performs correctly. Another advantage of using simulation to run policies is that simulation can run extensive cases to ensure extensive coverage. Thus, we can have both static and dynamic coverage, e.g., how many times a specific set of scenarios have run, and how many times a specific scenarios will happen, and how many of these scenarios are performed correctly.

24.5.2 Policy Specification

We designed the policy specification and enforcement language (PSEL) that covers obligation policies, authorization policies and system constraints. A policy editor

and a graphical policy-management interface have been developed for policy input. Obligation policies and authorization policies are defined on roles rather than on individual actors. A role represents a management position and the responsibilities and rights associated with that management position. Actors are assigned to roles according to their management positions. Since actors take particular roles in an organization, policies specified on a role will in turn apply to actors who take this role.

Obligation policies define a role's responsibilities, specifying what actions a role must or must not take under a condition. *Positive obligation policies* are event–condition–action (ECA) rules with the semantics that: on receiving a triggering event E, a role R must perform the action A if condition C is true. For instance, the policy "on receiving the call for fire, the supporting arms coordinator must issue a fire order" should be defined as a positive obligation policy, since it specifies the responsibility of actors who take the supporting arms coordinator role.

Negative obligation policies forbid a role from performing action A if condition C is true. For instance, the policy "If a main battle tank can shoot, it must not reject the fire order" should be defined as a negative obligation policy. If violated, the policy enforcer will inform the system simulator of the detection of the policy violation. The system simulator will perform the compensation action that is intended to minimize the consequences caused by the policy violation. Table 24.3 gives the syntax and examples of obligation policies.

Authorization policies define a role's rights to perform actions, specifying which actions are allowed or prohibited for the role under a certain circumstance. In PSEL, authorization policies are specified and enforced through access control models. Currently, two access control models are supported in PSEL: the Bell–LaPadula (BLP) model and the role-based access control model.

Bell–LaPadula (BLP) model [24.34] is a mandatory access control model widely used in military and government systems. It controls information flow and prevents information from being released to unauthorized persons. The BLP model defines four ordered *security levels*: Unclassified < Confidential < Secret < Top Secret. Security levels are then assigned to actors and data. An actor's security level is called the *security clearance*; data's security levels are called the *security classification*. Each action in the system has a subject (an actor) that performs the action, data (objects) on which the action is performed, and an accessing attribute that indicates the nature of this action (read, write, both, or neither).

The BLP model defines two access rules. The *no-read-up* rule applies to all actions whose accessing attributes are read. It specifies that an actor is not allowed to read data if the actor's security clearance is lower than the data's security classification. For instance, the observer in the special operations forces (SOF) team, with a security clearance of confidential, is not allowed to read (attribute: read) the target destroyed report (security classification: secret). The no-read-up rule prevents unauthorized persons from reading information they are not supposed to read. The *no-write-down* rule applies to all actions whose accessing attributes are write. It specifies that an actor is not allowed to write data if the actor's security clearance is higher than the data's security classification. The no-write-down rule prevents actors with higher security clearance from accidentally

Table 24.3 Examples of obligation policies

Policy type	Syntax	Example

Positive obligation policy	MUSTDO {
	definedOn ROLE
	triggeredBy EVENT
	do ACTION
	on CONDITION
	}
		MUSTDO {
		definedOn ROLE:SupportingArmsCoordinator
		triggeredByEVENT:ReceiveCFF
		do ACTION:IssueFireOrder
		on CONDITION:
		}

Negative obligation policy	MUSTNOTDO {
	definedOn ROLE
	do ACTION
	on CONDITION
	perform COMPENSATION
	}
		MUSTNOTDO {
		definedOn ROLE:MainBattleTank
		do ACTION:MBTRejectMission
		on CONDITION:MainBattleTankCanShoot
		perform COMPENSATION:Warning
		}

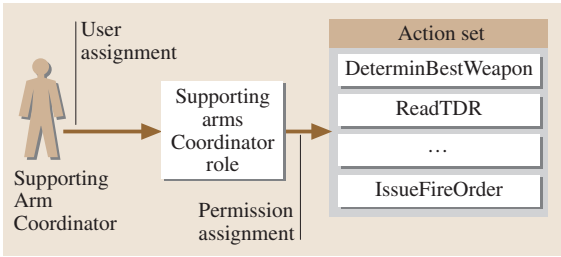


Fig. 24.15 The RBAC model

writing classified information to an unclassified media, so that unauthorized persons can read the information.

The *role-based access control (RBAC)* model [24.35–37] has been increasingly implemented in various systems, due to its policy-neutral nature. As shown in Fig. 24.15, in RBAC a group of roles are defined according to the semantics of a system. Actors are then assigned to roles according to their management position in this system. A set of actions are then assigned to a role, giving it the permissions to perform these actions. PSEL also supports a role hierarchy. *A role hierarchy* represents the superior and subordinate relationships among roles, allowing a superior role to obtain all permissions of its subordinate roles automatically. *Role delegation* is also supported, which enables a role to temporarily transfer its permissions to other roles, and for them to be revoked at a later time.

The access rule defined in the RBAC model is simple: an action A is allowed if

- there exists a role R, such that A.owner takes R, and
- R has permission to perform A.

For instance, the supporting arms coordinator (who takes the supporting arms coordinator role) is allowed to issue the fire order. Figure 24.16 shows the graphical user interface for role-based authorization policy specification.

System constraints define constraints on system status and behaviors that must hold in the system execution or simulation. *System constraints on data* specify that data must or must not be within a certain range. For example, the policy that "the distance between the SOF team and the surface-to-surface missile (SSM) launcher must be $\geq$ 3000 feet at all times" is a system constraint on data. *System constraints on actions* are currently temporal logic on actions. Examples could be "action DetermineBestWeapon must occur before action InputFireOrder" or "the call for fire (CFF) command can

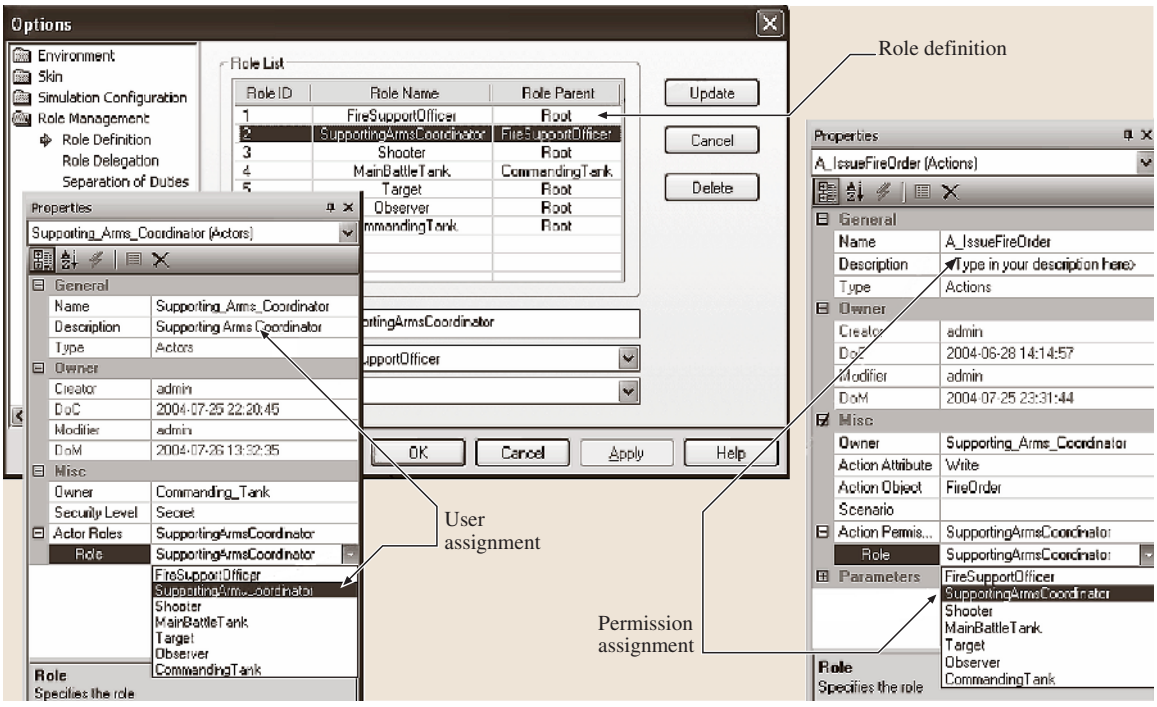


Fig. 24.16 Graphical user interface for role-based authorization-policy specification

Table 24.4 Examples of specifying system constraints

Policy type	Syntax	An example

System constraints on data	MUSTBE / MUSTNOTBE {
	appliedTo DATA
	status EXPRESSION
	on CONDITION
	perform COMPENSATION
	}
		MUSTBE {
		appledTo DATA:SOFTeamDistanceFromSSM
		status EXPRESSION:">=3000"
		on CONDITION:TRUE
		perform COMENSATION:SOFRetreat
		}

System constraints on actions	MUSTBE / MUSTNOTBE {
	do ACTION:Operator
	(action parameters)
	on CONDITION:TRUE
	perform COMPENSATION
	}
		MUSTBE {
		do ACTION:Sequence
		(ACTION:DetermineBestWeapon
		ACTION:InputFireOrder)
		on CONDITION:TRUE
		perform COMPENSATION:Warning }

be issued only once". Currently, the following temporal logic operators are supported by our ACDATE-based policy framework:

- Concurrency (A, B): A, B occur concurrently;
- Sequence (A, B, C): A, B, C occur in this sequence;
- Order (A, B, C): A, B, C occur in this order consecutively;
- Either (A, B): either A occurs or B occurs, but not both;
- Exist (A): A must occur;
- Once (A): A must occur once, and only once.

Conditions are associated with system constraints, specifying when these policies are to be enforced. In addition, compensation actions are defined in a system constraint. When policy violation is detected, associated compensation actions will be performed by the simulator to minimize the consequences brought about by policy violation. Table 24.4 gives syntax examples of specifying system constraints.

24.5.3 Policy Enforcement

Policies are enforced in the course of system simulation. In the initialization phase of system simulation, the policy enforcer will load policies out of the policy database and register them with the system simulator according to their semantics. Policies are registered so that the system simulator knows when to trigger policy enforcement. The system simulator triggers the policy enforcer when a registered event occurs, a registered action is performed, or a registered datum is modified. The policy enforcer will enforce relevant policies attached to these registered events, actions, or data, and return the results of enforcement back to the system simulator.

Policy enforcement can be classified into three categories: policy checking, policy execution, and policy compensation. *Policy checking* verifies if policy violations are detected when actions are performed or data are changed. *Policy execution* executes the action defined in the policy when receiving the triggering events. *Policy compensation* executes the compensation action defined in the policy when policy checking detects a policy violation. All checkable policies come with a compensation action.

Only positive obligation policies are executable policies. The other types of policies (e.g. negative obligation policies, all authorization policies and all system constraints) are all checkable policies. *Executable policies* influence the paths of system simulation through the actions defined in them. When the triggering event occurs, executable policies registered to this event will be enforced, and the action specified in the policy specification will be executed by the system simulator. *Checkable*

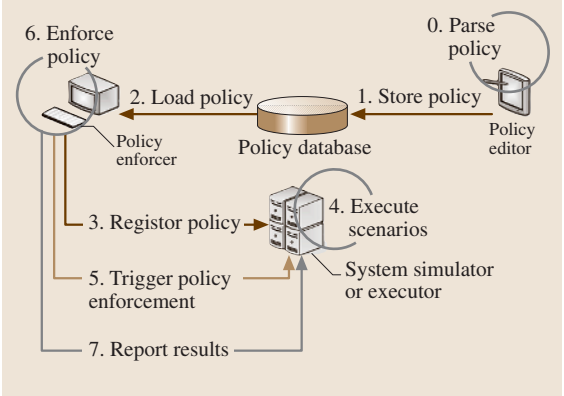


Fig. 24.17 The policy enforcement framework

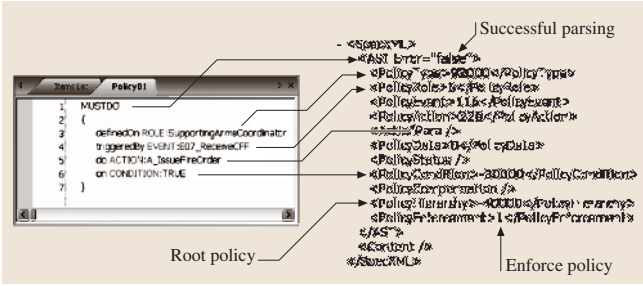


Fig. 24.18 Policy parsing

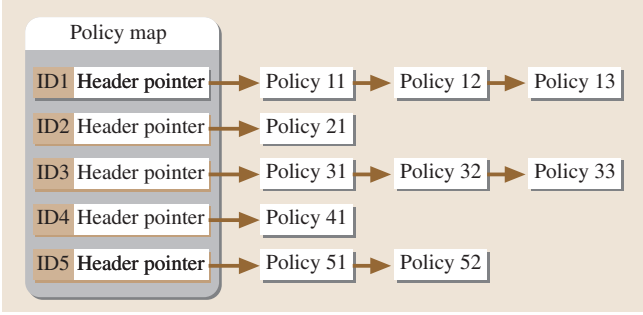


Fig. 24.19 Policy map

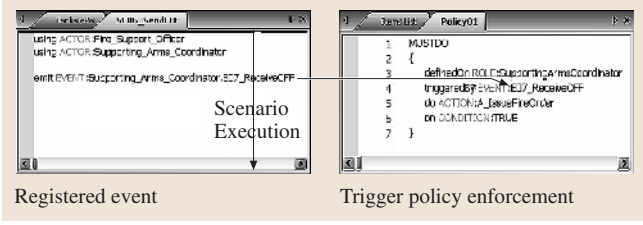


Fig. 24.20 Policy enforcement triggering – event

policies can also influence the paths of the system simulation through compensation actions. When data are changed or actions are performed, checkable policies registered to the data or actions will be enforced. If policy violations are detected, the compensation action specified in the policy specification will be executed by the system simulator.

Figure 24.17 illustrates the policy enforcement framework. After ACDATE elements are defined and policies are extracted, policies are specified in the policy editor, which parses policies for correctness, C&C, and stores them in the policy database. During the initialization phase of simulation, the policy enforcer loads policies from the policy database, interprets them and registers them with the system simulator. While the simulator executes system scenarios, it triggers the policy enforcement when registered events occur, registered actions are performed, or registered data are changed. The policy enforcer checks or executes policies and returns the results to the simulator. Based on the returned results, the simulator determines what the next system scenarios are.

The policy editor parses policies for correctness and C&C. On successful parsing, the policy editor translates policies into XML and stores them in the policy database. The policy parser is implemented by: another tool for language recognition (ANTLR). According to the policy syntax, ANTLR creates an abstract syntax tree (AST) for each policy. Policy elements (roles, actions, condition, etc.) are extracted by traversing the tree, and translated to the XML representation, as shown below. The XML representation of a policy is then stored in the policy database as a string. Figure 24.18 shows an example policy parsing.

The purpose of policy registration is to let the system simulator know when policy enforcement should be triggered. In our ACDATE-based policy framework, three out of the six ACDATE elements can trigger policy enforcement: event, action, and data. Events occurrences will trigger the enforcement of positive obligation policies; action performances will trigger the enforcement of negative obligation polices, authorization policies, and system constraints on actions; data changes will trigger the enforcement of system constraints on data, as shown in Table 24.5.

To improve performance, a policy map is created, mapping a particular event, action or data to a list of relevant policies to which it is registered, as shown in Fig. 24.19. All policies registered to a particular action,

Table 24.5 Policy registration

Policy type	Authorization policy	Positive obligation policy	Negative obligation	System constraints on data	System constraints on action
Event		X			
Action	X		X		X
Data				X	

event or datum form a linked list. The linked list and the identification (ID) of the action, event or datum are then organized as a policy map. When policy enforcement is triggered, the policy enforcer locates the policy linked list of a particular action, event or datum, and enforces all policies in the list.

After policies have been registered, the simulator initializes the data and starts running the system scenarios. The simulator keeps an eye on the simulation of system scenarios, and triggers the policy enforcement by invoking the EnforcePolicy method in the policy enforcer whenever an event is triggered, an action is performed or a datum is changed. Figure 24.20 shows an example where a triggering event causes a policy enforcement execution. The policy enforcer enforces all relevant policies, records all violations in the policy log, and returns the policy log back to the system simulator.

When policy enforcement is triggered, the simulator invokes the EnforcePolicy method in the policy enforcer, passing the ID of the event, action or datum that triggered the policy enforcement. On being triggered, the policy enforcer looks into its policy map, maps the ID to a list of policies to which it has been registered, and enforces them one by one. Authorization policies are not registered in the policy map, and they are enforced before obligation policies and system constraints are enforced. When policies are being

```
bool CPolicyEnforcer::PolicyEnforcer(const long lEntityID)
{
    // lEntityID is the ID of the entity that triggers policy enforcement
    // An Entity could be an event, action, or data
    if (GetEntityType(lEntityID) == ACTION_TYPE)
    {
        Verify whether this action is allowed by RLP  Model;
        Verify whether this action is allowed by RBAC Model;
    }

    deque<CPolicy*> pDequePolicy = m_mapPolicyMap.Find(lEntityID);
    for (size_t i = 0; i < pDequePolicy.size(); i++)
    {
        bool bPassed = true;
        CPolicy* pPolicy = pDequePolicy[i];

        switch(pPolicy->m_lPolicyType)
        {
        case POLICY_MUST_DO:
            bPassed = PolicyCheckMustDo(pPolicy, lEntityID);
            break;
        case POLICY_MUST_NOT_DO:
            bPassed = PolicyCheckMustNotDo(pPolicy, lEntityID);
            break;
        case POLICY_MUST_BE:
            bPassed = PolicyCheckMustBe(pPolicy, lEntityID);
            break;
        case POLICY_MUST_NOT_BE:
            bPassed = PolicyCheckMustNotBe(pPolicy, lEntityID);
            break;
        }

        if (!bPassed)
        {
            Record Policy Violations;
        }
    }

    if (no policy violations) return true;
    return false;
}
```

Enforce authorization policies

Enforce obligation policies

Enforce system constraints

Fig. 24.21 Algorithm for policy enforcement

enforced, all violations are recorded into a policy log that is returned to the simulator. The EnforcePolicy method returns a Boolean value indicating whether policy violations are detected. Figure 24.21 gives the policy enforcement algorithm.

24.6 Dynamic Reliability Evaluation

Software reliability has been defined as the probability that no failure occurs in a specified environment during a specified (continued) exposure period. Existing software reliability models assess reliability statically in the development process. The E2E T&E perform dynamic evaluation at runtime using a software reliability model that is integrated into the ACDATE scenario model. Figure 24.22 illustrates the development and operation processes using this model.

From the scenario specification, the atomic components can be identified and a data collector is instrumented around each atomic component, which collects runtime failure data during testing and operation. Based on the collected data, the reliability of both components and the SoS can be assessed. The rest of the section explains the major components in this reliability assurance process.

24.6.1 Data Collection and Fault Model

An SoS may consist of many subsystems or components and it is hard to exactly distinguish their contributions to the overall reliability of the SoS due to the anfractuous dependency relations among them. A component may consist of several subcomponents, in which case its reliability can be computed analytically, provided the reliability of each subcomponent is known. The breakdown can be continued to each subcomponent. However it must end somewhere when the component is either indivisible or it is not worthwhile dividing it further. We then consider these components as black boxes or atomic components in our reliability model. In other words, the reliability of an atomic component is not the result of (but an input to) our reliability model.

Although the decision on what component shall be treated as a black box is truly application-dependent,

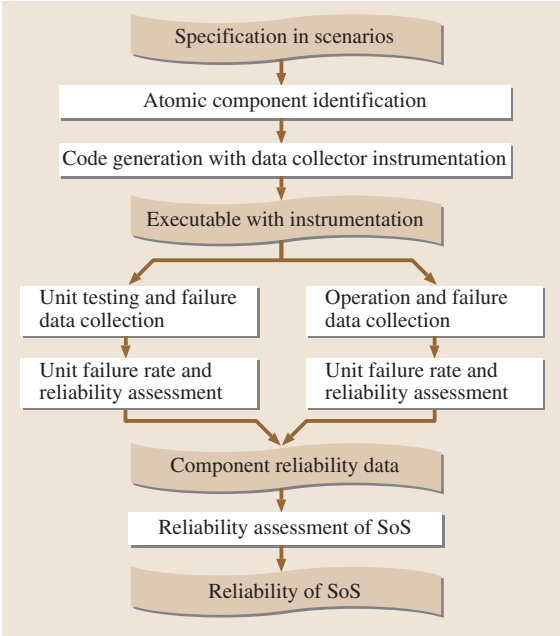


Fig. 24.22 Dynamic reliability assurance

we propose three general principles to curb arbitrariness and strengthen the rationale for our reliability model.

Granularity principle: a finer granularity can lead to more accurate evaluation results but may increase the complexity of the computation. Basically there is a trade-off between granularity and accuracy and their balance depends on the specific application and its requirements.

Perceptible principle: if a component is treated as a *white box*, the opposite of a black box, then the dependencies among all its subcomponents that have an effect on reliability will be modeled explicitly and hence be accounted for during the reliability computation. If the dependency is neither clear nor completely modeled, then the result of the computation will be biased.

Continuous principle: a component is a white box only if its super-component is a white box. It is of no benefit if a black box has a subcomponent as a white box since the reliability of a black box is not computed.

The perceptible principle and the continuous principle define the *effective domain* of the ACDATE scenario model. Inside the domain, everything is explicitly modeled and hence is a white box, while a component outside the domain is a black box. Code that is either manually developed or automated generated based on the model represents its effective domain in the system.

The black box, which is outside the effective domain of the model, can be further categorized as follows:

- Operation on hardware through a device driver;
- A system call provided by the operating system;
- A method or attribute in the programming platform, e.g., the vector class in Java, C# or C++ standard template library (STL);
- A method or attribute in a library provided by a third party;
- Input from a human operator;
- A component in a remote location.

Different types of black boxes incur different reliability estimations, which will be detailed in the next subsection.

Figure 24.23 illustrates the effective domain, where a circle is a white box, a disk is a black box, a line is a dependency relation, and a dashed line is an unclear dependency relation.

To collect failure data, each function call to an atomic component is replaced by a wrapper call that collects failure data related to the atomic component.

Assume a call to an atomic component is atomfun(p1, p2, . . . , pn), where atomfun is the function name, and p1, p2, . . . , pn are the parameters. In the data collector instrumentation stage in Fig. 24.22, the function call is replaced by a wrapper function call: dataanalyzer(atomfun, p1, p2, . . . , pn).

The data collector function is:

```
dataanalyzer(atomfun, p1, p2, . . . , pn)
{
Increment the execution counter;
Find the specification of atomfun;
Verify the legitimacy of p1, p2, . . . , pn;
Call atomfun(p1, p2, . . . , pn);
```

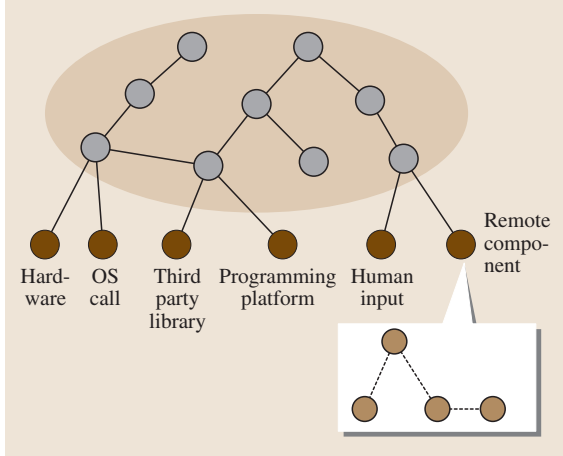


Fig. 24.23 Identification of atomic components in an SoS

Verify the legitimacy of results;
Handle exceptions;
If results fail, report a failure;
}

The data collector performs acceptance testing on the inputs and outputs of the atomic component and maintains following data in a local log file:

- The execution counter keeps track of how many times the atomic component has been called.
- The failure counter counts how many failures have been detected.
- Failure types: incorrect data, exceptions, and crash, etc.

The local log file associated with the data collector automatically synchronizes with a central database that records data related to all atomic components. The synchronization can be performed in the testing stage, in the development process, and during the operational stage after the software is delivered to the client. Online error reporting during the operational stage can also help the developers to design patches and updates of the software. The execution number of each atomic component collected during the operational stage can be used to determine the execution profile of the software.

For each atomic component, the following data items are maintained in the database:

- Versions: recording the date and time of each modification/error correction performed on the atomic component. Each error correction results in a new version of the component;
- Number of executions between two error corrections;
- Number of failures between two error corrections;
- Numbers of each failure type.

Table 24.6 Reliability definition of ACDATE entities

ACDATE	Reliability of the ACDATE entity
	The *probability* that the:
Actor	– actor presents the expected behavior
Condition	– condition presents the expected Boolean value
Data	– data presents the expected value
Action	– action presents the expected behavior
Timing	– action completes in given time frame
Event	– event is sent or received successfully
Scenario	– scenario presents the expected behavior
System	– system presents the expected behavior

These data can be used to estimate the reliability of the components. In the next subsection, we will apply the input domain-based reliability growth model to estimate the reliability of each atomic component.

An incorrect output of a program is a failure. A program contains errors if it can produce a failure when certain input cases are applied. The size of an error is the ratio of the number of inputs that can detect the error (cause a failure) and the total number of valid inputs.

According to the input domain-based reliability growth model [24.38, 39], the failure data stored in the central database can be used to estimate the error sizes $\Theta_1, \Theta_2, \ldots, \Theta_k$ and the failure rates $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the errors in each atomic component between two error corrections. The error sizes and failure rates between error corrections can be used to estimate the final failure rate λ of each atomic component.

The failure rate and the total number of executions associated with each atomic component is the input to the structural reliability model to be discussed in the remaining part of the paper.

24.6.2 The Architecture-Based Reliability Model

In the previous subsection, we evaluated the reliability of atomic components. In this subsection, we evaluate the reliability of an SoS consisting of multiple systems, each of which is considered an atomic component. The model can be generalized to evaluate a system or a subsystem in a system, with the knowledge of the reliability of its components, operational profile, and the architecture of the system. The architecture determines the contribution of the reliability of each atomic component to that of the overall system. Hence the approach is named the architecture-based reliability model [24.40].

First we give the definition of a component's reliability and present our assumptions; then we discuss how the architecture affects the propagation of reliability; finally, we derive the formulas that compute the reliability.

We base our reliability model on the ACDATE scenario model, which describes the structure of a system using model entities *actors*, *conditions*, *data*, *actions*, *timing*, and *events*, and the behavior of a system using *scenarios*. The ACDATE scenario model models the general computing process.

The *reliability* definitions of ACDATE entities and scenarios are summarized in Table 24.6.

The assumptions of our reliability are

1. Assignment assumption: the assignment operation introduces no new failure.
2. Condition assumption: the condition fails when any data that constitutes the condition fails.
3. Acyclic dependency assumption: there is no cyclic dependency among ACDATE entities.

The system behaviors are specified by a few system-level scenarios and the reliabilities of those system-level scenarios will contribute to that of the system. Different scenarios may have different execution rates (the operational profile), which determine the weight of the contributions.

A scenario is a sequence of activities connected by four operators: *sequence*, *choice*, *loop* and *concurrency*. Each *activity* is a data assignment, exchanging an event, doing an action, or executing a sub-scenario. Hence the reliability of data, events, actions and sub-scenarios will contribute to that of a scenario. The choice and loop operator are associated with one or more conditions that determine the branches to take. Hence the reliability of conditions also contributes to that of a scenario. Moreover, the *true/false* rate of each condition will affect the reliability of the scenario through the choice and loop operators (formulas will be presented later).

Each top-level scenario would be invoked by an *external* event. Hence, the occurrence rates of external events determine the operational profile of top-level scenarios. A scenario may emit an *internal* event, whose sole function is to resume or invoke the execution of a sub-scenario. Hence, the occurrence rates of internal events will affect the operational profile of sub-scenarios (in addition to direct calling from other scenarios). The occurrence rates of internal events can be determined by that of external events invoking top-level scenarios, and the control flow of those scenarios that emit internal events.

To summarize, assuming that we know the reliability of each scenario and the occurrence rate of each external event (and hence internal events), we can evaluate the reliability of the system following the formula:

$$Rel_{\text{system}} = \left[\sum (w_i * Rel_{\text{scenario\_}i})\right] / \sum (w_i) ,$$

where w_i is the execution rate of the corresponding scenario. In the following we present the calculation of the reliability of a scenario.

The reliability of data is determined by that of its storage method, which is modeled as atomic components (memory, external database, file system, etc.), and hence is known. The reliability of actions is known if it is atomic (e.g., a system call), or is that of the sub-scenario that implements it. The reliability of events is determined by that of the communication link (atomic component) and hence is known.

Following assumption 1, the reliability of assignment is that of the right-hand-side data. Hence we know the reliability of each activity in a scenario.

If several activities are connected by a sequence operator, then the overall reliability follows the formula:

$$Rel_{\text{sequence}} = \prod Rel_{\text{activity\_}i} ,$$

where $Rel_{\text{activity\_}i}$ is the reliability of each activity that participates in the sequence. If several activities are connected by a concurrency operator and all of them are replicas, then the overall reliability follows the formula:

$$Rel_{\text{concurrency}} = 1 - \prod (1 - Rel_{\text{activity\_}i}) .$$

Otherwise, it is the same as the sequence, since any failure results in the failure of the overall concurrency. For the loop operator, the formula is:

$$Rel_{\text{loop}} = (Rel_{\text{cond\_set}} \cdot Rel_{\text{block}})^{Pt} ,$$

where $Rel_{\text{cond\_set}}$ is the reliability of the condition set associated with the loop operator, Rel_{block} is the reliability of the block of activities enclosed in the loop operator, and Pt is the expected number of loops. We will discuss $Rel_{\text{cond\_set}}$ later. For the choice operator, the formula is

$$Rel_{\text{choice}} = Rel_{\text{cond}_{\text{set}}} \cdot \left(Pt \cdot Rel_{\text{true\_block}} + (1 - Pt) \cdot Rel_{\text{false\_block}}\right) ,$$

where Pt is the probability that the condition set evaluates as *true*. Since a scenario consists of only these four types of operators, we can calculate its reliability following these formulas. The reliability of a condition is determined by the data that constitute the condition, or is known if the condition is atomic (e.g., a system call). Following assumption 2, if a condition consists of several data, its reliability is the product of the reliabilities of all the data. We omit the deduction process due to the space restriction and only present the final formulas here:

$$Rel_{\text{cond\_set}} = 1 - \sum ProTop_{c(m,o)} ,$$

where each $ProTop_{c(m,o)} = \sum ProTop_{\{c\}o}$

Table 24.7 The most reliable services and their forecast

Components	Reliability	Forecast probability	Adjusted probability
RainForecast	0.764	18 % heavy rain	33.1%
TempForecast	0.98	31 % extreme temp	31.76 %
WindForecast	0.90	23 % strong wind	28.4 %

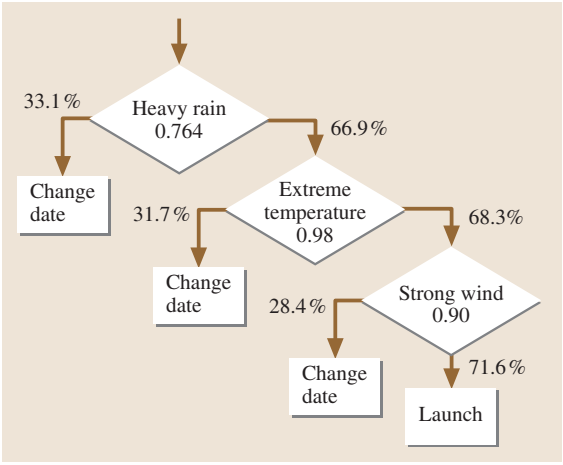


Fig. 24.24 Decision making

Each $ProTop_{\{c\}o}$ is calculated by the following formulas:

$$ProTop_{\{c\}o} = Rel_{\{c\}o} \cdot \left(ProTF_{\{c\}o} + ProFT_{\{c\}o}\right) \cdot Prob_{remv} ,$$

$$Rel_{\{c\}o} = \prod(1 - Rel_{Ck}) \cdot \prod(Rel_{Cl}) ,$$

where Rel_{Ci} is the reliability of the i-th condition in the condition set. Each condition set is evaluated in disjunction normal form (DNF), whose Boolean value may be dominated by a true disjunct. $Prob_{remv}$ is used to compensate for possible domination and is defined as the probability that the disjunct evaluates to be false. $ProTF_{\{c\}o}$ and $ProFT_{\{c\}o}$ are the probabilities that a condition incorrectly changes from *true* to *false*, or from *false* to *true*, respectively, and are determined by the condition's reliability.

24.6.3 Applications

This subsection uses an example to illustrate the applications of the proposed dynamic software reliability model. Assume a space agency plans to launch a satellite on a specific date and from a specific location. Among other constraints, the launch is heavily dependent on the weather conditions at the launch location, including rain, wind, and temperature. Three independent weather services are used, offering RainForecast, TempForecast, and WindForecast, respectively. The forecasts are given with their probabilities. The reliabilities based on the history of the services, their forecast probabilities (component outputs), and the adjusted probabilities based on the reliability and the forecast probabilities are given in Table 24.7.

To decide whether to change the launch date based on the weather forecasting information, the space agency then constructed a system based on the components. The reliability of the system and the final decision of whether to launch the satellite can then be assessed by the process shown in Fig. 24.24. The numbers in the diamond boxes are the reliabilities of each component. The numbers on the branches are the probabilities forecasted by the best service. The decision is based on these two factors.

24.6.4 Design-of-Experiment Analysis

Design of experiment (DOE) is an engineering technique [24.41] that can be used to determine the extent of the impact of the parameters (factors) of a model on the final results. This subsection applies DOE to analyze the impact of the reliability of the components on the reliability of the SoS.

There are three factors in the example, the reliabilities of (A) RainForecast, (B) TempForecast, and (C) WindForecast. We use two-level DOE techniques, i. e., we use high and low values of each factor: RainForecast (70%, 90%), TempForecast (90%, 99%) and WindForecast (85%, 95%). In our experiment, the three-factor and two-level design generated the analysis of variance (ANOVA) table shown in Table 24.8.

The F-value represents the significance of the impact of a model and its components. In general, if a com-

Table 24.8 ANOVA significance analysis

Source	F-value	Prob > F-value
Model	$1.421 \times 10^{+5}$	< 0.0001
A (RainForecast)	$3.898 \times 10^{+5}$	< 0.0001
B (TempForecast)	$2.2943 \times 10^{+4}$	< 0.0001
C (WindForecast)	$1.3558 \times 10^{+4}$	< 0.0001

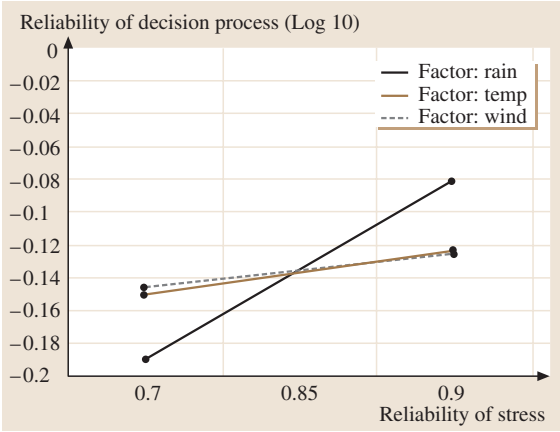


Fig. 24.25 Impact of components

ponet generates a significance value (Prob > F-Value) of less than 0.05, the impact of the component is significant. For example, the F-value $1.421 \times 10^{+5}$ for the overall model in the table implies that the model is significant. There is only a 0.01% chance that the an F-value of this size could occur for this model due to noise.

The experimental results in Table 24.8 also show that the F-values and significances of RainForecast, TempForecast, and WindForecast are all less than 0.0001, and thus they are all significant model components.

DOE can be used to compare the significances among the components. Figure 24.25 shows the impacts of the three components on the overall reliability in our example. As can be seen, the higher the component reliability, the higher the overall reliability. However, the impact of the RainForecast service is much more significant than that of the others. This suggests that the space agency should pay more attention to the quality of the rain-forecast service provider.

24.7 The Fourth Generation of E2E T&E on Service-Oriented Architecture

Service-oriented architecture (SOA) and web services (WS) are emerging technologies that may change the way computer software is designed and used. Many industrial standards have been defined in the past few years to facilitate and regulate the development of WS. However, there are still a number of barriers preventing WS from being widely applied or being used as the platform for trustworthy and high-assurance systems. *Sleeper* identified five missing pieces of WS technology: reliability, security, orchestration, legacy support, and semantics [24.42]. Among these five issues, reliability is the least addressed and probably most difficult, for the following reasons:

- WS are based on an unreliable and open internet infrastructure, yet they are expected to be trustworthy.
- WS have a loosely coupled architecture, yet they are expected to collaborate closely and seamlessly.
- WS can be invoked by unknown parties with unpredictable requests, and thus WS must be robust.
- WS involve runtime discovery, dynamic binding with multiple parties, including middleware and other WS, and runtime composition using existing WS. Thus, WS must support dynamic and runtime behaviors.
- WS must support dynamic configuration and reconfiguration to support fault-tolerant computing.
- WS must support dynamic composition and recomposition to cope with the changing environment and changing requirements.
- WS involve concurrent threads and object sharing. It is difficult to test concurrent processes.

We propose an integrated collaborative and cooperative WS development process to achieve high-assurance computing. The process is implemented in a framework consisting of three major modules dealing with the construction, publishing, and testing of WS.

As shown in Fig. 24.26, the WS cooperative and collaborative computing (WSC3) framework consists of three modules: cooperative WS construction; publishing; and testing, assessment, and ranking (WebStrar). The framework can be used by service requestors, service providers, as well as researchers experimenting with WS.

As an example, the arrows and numbers in Fig. 24.26, which outline cooperation scenarios between the components, are explained as follow.

1. The WS construction module, in the process of dynamically constructing a composite WS based on existing WS, requests information from the WS publishing module.
2. The publishing module provides required information, including the specification and interface of using the WS.

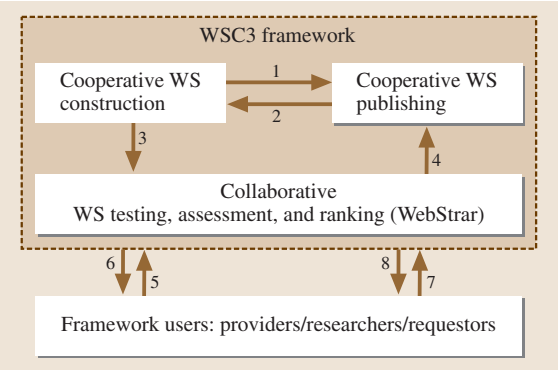


Fig. 24.26 WS cooperative and collaborative computing framework

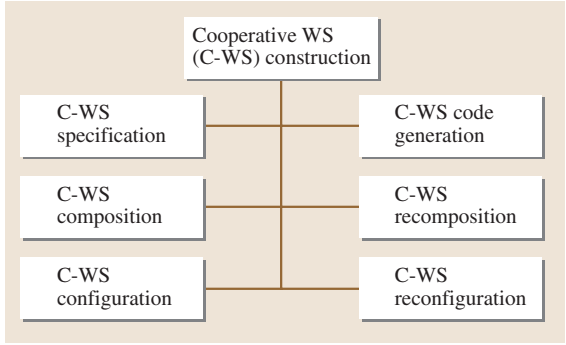


Fig. 24.27 Cooperative WS construction

3. After a composite WS is constructed, the construction module submits the WS to the WebStrar module for rigorous testing.
4. If the WS passes the test, it will be registered with the publishing module and a new WS is available for online access.
5. A WS provider or a researcher submits their WS for publication or testing. The WS will be tested by the WebStrar module rigorously based on the test scripts submitted by the provider as well as the test scripts generated by WebStrar. Sharing the test scripts represents collaboration between the framework and WS providers and researchers.
6. The framework publishes the WS and informs the WS provider if the submitted WS passes the test.
7. A WS requestor requests a service. The requestor can request testing before using a WS. It can use the test scripts provided by the framework or submit their own test scripts. Sharing the test scripts represents collaboration between the framework and WS requestors. WS requestors can also access the reliability data and ranking information of published WS.
8. The framework processes and responds to the WS requestor.

In the following three subsections, we elaborate the three modules in the WSC3 framework, respectively.

24.7.1 Cooperative WS Construction

Figure 24.27 elaborates the cooperative WS construction module in Fig. 24.26. This module has six components.

The cooperative WS specification component provides guidelines and tools for users to write WS specifications in a specification language, e.g., in OWL-S. The component will then use WebStrar to perform a consistency and completeness (C&C) check on the specification.

Once the specification passes the check, the code-generation component can automatically generate the executable code. The WS generated in this way is atomic, because its implementation detail is not accessible to the users. All WS submitted by WS providers are also atomic.

The cooperative WS composition component provides automated high-level WS composition based on existing atomic WS and their specifications.

The recomposition component can reconstruct a composite WS if the requirement and specification are changed. Composition and recomposition components construct WS based on the functional requirement while the configuration and reconfiguration components deal with the management of redundant resources and the reliability of WS. The configuration component adds redundant structure into composite WS to meet the reliability requirements while the reconfiguration component maintains redundancy after the environment is changed, for example, if some WS become faulty or unavailable.

24.7.2 Cooperative WS Publishing and Ontology

This section elaborates the cooperative WS publishing module in Fig. 24.26. Current WS publishing is based on the universal description, discovery, and integration (UDDI) technique. The UDDI discovery part is based on simple term/text matching, which does not have the intelligent to find synonyms and semantically related terms. For example, if the phrase "red wine" is searched, terms like Cabernet Sauvignon and Merlot should be found too.

Table 24.9 Cooperative versus traditional ontology

	Traditional ontology	Cooperative ontology
Test scripts	Does not include test scripts	Include test scripts and execute these test scripts at runtime
Nonfunctional property	Use certain terms to represent the value of these nonfunctional properties, such as performance and security.	Use test scripts to present the nonfunctional properties. Translate the nonfunctional properties to the measurable features.
Behavior constraints	Allow specification of the constraints, but not their execution at runtime	Execute the constraints at runtime to check if the assigned services match the constraints.
Interface	Does not include the interface information in the description	Use specific test scripts to present the interface constraints

OWL-S and Protégé are recent projects that support ontology description. Ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary [24.43]. This represents knowledge about a domain and describes specific situations in the domain [24.44]. Dynamic composition and recomposition need to discover WS at runtime over the internet, add them to the ontology domain, and then compose a WS at runtime.

Current ontology methods do not include the verification process. In our cooperative WS publishing module, we integrated the collaborative verification and validation (CV&V) process into the ontology by including the necessary test scripts in the ontology domain. When a WS is chosen for composition, recomposition, configuration, or reconfiguration, the stored test scripts will be immediately applied to test the WS. This integrated ontology with CV&V is called cooperative ontology. Table 24.9 compares and contrasts cooperative and traditional ontology.

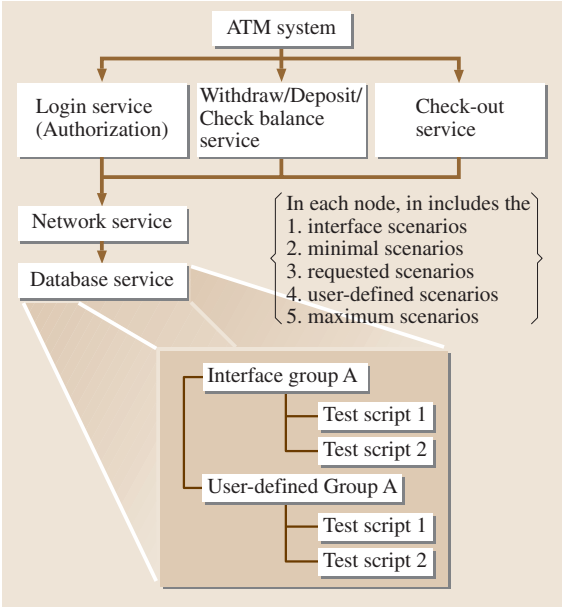


Fig. 24.28 ATM services tree example

The ontology-based architecture plays a key role in runtime WS composition. Runtime verification can choose different levels of test scripts to verify the services found. To further explain the idea, a simple automatic teller machine (ATM) example is used here. Assume the ATM offers login, balance-checking, withdrawal, deposit, and logout services.

The service tree of the cooperative ontology representing the ATM composite WS is given in Fig. 24.28. Each node has a service interface definition, a number of service constraints and service scenarios, and user-specific requirements described using test scripts. For instance, if we want to choose a login service that supports a specific character set [&*^%] in the user name, the user-defined test script can be:

Execute Register("abc&^%123", "123456");*

If this test script fails, the services found do not conform to the requirement and will be rejected. As discussed before, there are multiple levels of test scripts, which include the interface test scripts. In the service tree specification, the internal relations among test scripts are also important to support dynamic service composition and recomposition.

24.7.3 Collaborative Testing and Evaluation

The WS composed using the process in Fig. 24.29 will be tested, assessed and ranked. Figure 24.4 depicts the module that tests and assesses the reliability of the WS and assures the tools involved. The solid arrows indicate that a component can be decomposed into several subcomponents, while the dotted arrows indicate the data flow between components. WebStrar itself is a framework supporting the development of trustworthy WS

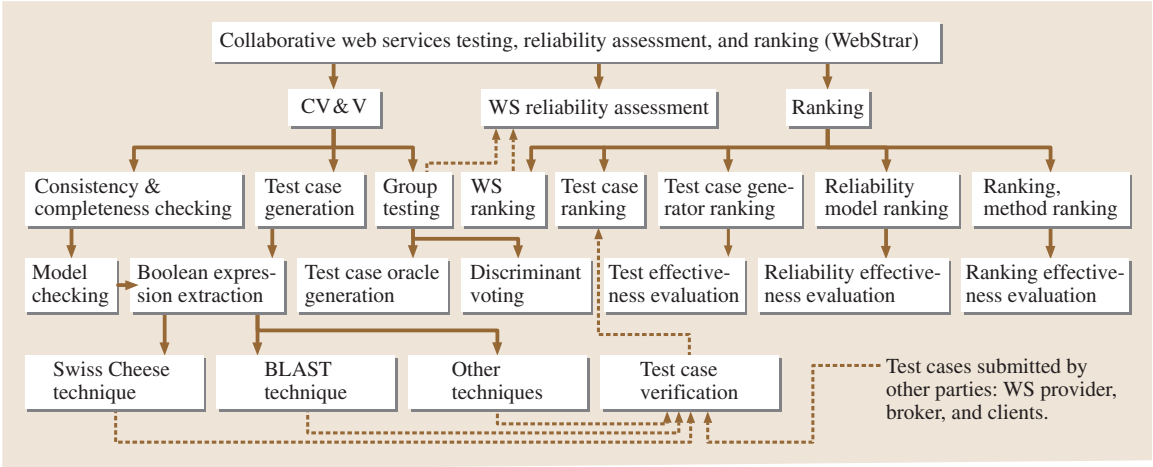


Fig. 24.29 Collaborative testing, assessing, and ranking

(http://asusrl.eas.asu.edu/srlab/projects/webstrar/index.htm). This section explains the individual techniques developed and to be developed in this framework. At the top level, WebStrar consists of three components: CV&V, WS reliability assessment, and ranking.

The idea of CV&V is to involve all parties (WS providers, brokers, and clients) in verifying and validating the WS, because the WS provider does not have full information on how their WS will be used by clients and brokers in user-composed composite WS [24.45, 46]. Before test-script generation, the consistency and completeness of the WS specification in OWL-S or web services description language (WSDL) will be checked through model checking or other methods, which may detect inconsistent conditions or incomplete coverage of the requirements [24.17]. Once the specification passes this check, Boolean expressions can be extracted, which can be used for test-script generation. Different techniques can be applied here; we have applied the Swiss cheese [24.17] and BLAST [24.16] techniques in our experiments. The system-generated test scripts, along with the test scripts provided by other parties, will be verified for correctness and ranked according to their effectiveness at detecting faults in WS.

Group testing is a key technique developed to test the potentially large numbers of WS available on the internet [24.46]. WS with the same specification can be tested in groups and the results are compared by a discriminant voter, which can identify correct and faulty output based on the majority principle. The majority results are used as the oracles for future testing.

Reliability assessment of WS is different from that of traditional software. WS reliability models do not have access to the WS source code. WS reliability can only be assessed at runtime because WS can be composed and modified (recomposed) at runtime. A group-testing-based dynamic reliability model has been developed to assess WS reliability [24.40]. Reliability is one of the criteria used to rank WS. Other criteria include security, performance, real-time ability, etc.

WebStrar allows researchers and WS providers to submit their models and tools for evaluation and ranking. WebStrar supports ranking of test scripts in terms of fault-detection capacity, test-script generation algorithms in terms of generation of effective test scripts, reliability models in terms of the accuracy of their assessment results, and ranking models themselves in terms of ranking accuracy.

24.8 Conclusion and Summary

E2E T&E technology was initially developed for DoD command-and-control systems and later applied in various industrial projects. It was initially designed as an integrated testing technology and later developed

into a full development technology spanning requirements; specification; model checking; code generation; test-case generation; testing; simulation; policy specification and enforcement; and reliability, security, and risk enforcement and assessment. All these techniques are coherently based on the scenario specification. From the software architecture point of view, E2E T&E technology has evolved from centralized architecture, through distributed agent architecture, to service-oriented architecture.

E2E T&E technology has been successfully applied in several DoD command-and-control projects where high-assurance computing is required and in several civilian projects including embedded systems in business networks. The application of the technology and its tools dramatically reduce the development cycle of the end systems and increase their dependability.

Systems developed using E2E T&E technology have the following attributes and features.

The developed systems are flexible and can adapt to changing environments. Some important attributes of *adaptability* include *speed*, *scalability*, *reusability*, *partitioning*, and *integration*. In general, the system is adaptive as it supports rapid development; is scalable from small applications to large applications; has many reusable tools that can produce reusable components; and has an integrated process.

The E2E T&E development process is fast because it includes tools that perform all jobs automatically where possible. It generates test cases automatically from the system specification and policy specification. It generates an executable automatically, and performs distributed test execution automatically. The evaluation process is also automatic.

The E2E T&E process is scalable because it can apply to large as well as small applications. The scenarios used in the E2E process are hierarchical and thus can apply to the hierarchical structure of a large SoS or a small subsystem.

The E2E T&E process has many reusable tools, including the scenario specification tool, the test-case management tool, the scenario simulation tool, and the distributed test-execution tool. One of the key benefits of the E2E tool is that specified scenarios are highly reusable and can be easily changed. System scenarios keep on changing as new requirements become known and new technology is introduced during system development; changing system scenarios with the E2E tool is much easier than redeveloping scenarios by hand. In most cases, new scenarios are developed by changing existing scenarios, and changing scenarios using the E2E tool with dependency analysis is easier than starting from scratch without any tool support. Furthermore, once new scenarios are specified, they can be automatically analyzed by various techniques such as timing analysis, and new test scripts can be rapidly generated and executed.

E2E T&E tightly integrates system analysis and modeling with integration testing because the same techniques, i. e., scenarios, can be used for both system analysis as well as integration testing. The importance of testing has recently being emphasized by agile development processes such as extreme programming. While testing is one of their main techniques, agile processes do not have such tight integration between system analysis and testing as the DoD E2E T&E. Tight integration changes the way systems are developed; instead of performing *requirement-driven testing* only, the E2E process calls for *test-based requirement analysis*. In other words, the requirements should be developed in a way that can be used for rapid integration testing (by automated test-script generation, verification patterns, and distributed test execution) and evaluation (by various analyses and simulation). In fact, E2E T&E supports a test-based development process from requirements to operation and maintenance, and such a process is compatible with agile development processes or incremental development.

References

24.1 R. S. Pressman: *Software Engineering: A Practitioner's Approach*, 5th edn. (McGraw Hill, New York 2000)

24.2 S. Kirani, W. T. Tsai: *Specification and Verification of Object-Oriented Programs*, tech. rep., Department of Computer Science and Engineering (Univ. Minnesota, Minneapolis 1994)

24.3 D. C. Kung, P. Hsia, J. Gao: *Testing Object-Oriented Software* (IEEE Computer Society, Los Alamitos, CA 1999)

24.4 W. T. Tsai, Y. Tu, W. Shao, E. Ebner: Testing extensible design patterns in object-oriented frameworks through hierarchical scenario templates, Proc. COMPSAC **23**, 166–171 (1999)

24.5 W.T. Tsai, V. Agarwal, B. Huang, R. Paul: Augmenting Sequence Constraints in Z and its Application to Testing. In: *Proc. 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology* (IEEE Computer Society Press, Los Alamitos 2000) pp. 41–48

24.6 DoD OASD C3I Investment, Acquisition: *End-to-End Integration Testing Guidebook* (IEEE Computer Society Press, Los Alamitos 2001)

24.7 W.T. Tsai, X. Bai, R. Paul, L. Yu: *Scenario-Based Functional Regression Testing* (IEEE Proc. of COMPSAC, Chicago 2001) pp. 496–501

24.8 W.T. Tsai, C. Fan, R. Paul, L. Yu: *Automated Event Tree Analysis Based-on Scenario Specifications* (Proc. of IEEE ISSRE, IEEE Computer Society Press, Los Alamitos 2003) pp. 240–241

24.9 W.T. Tsai, X. Bai, R.J. Paul , W. Shao, V. Agarwal: *End-To-End Integration Testing Design* (Proc. of IEEE COMPSAC, Chicago 2001) pp. 166–171

24.10 X. Bai, W.T. Tsai, R. Paul, K. Feng, L. Yu: *Scenario-Based Modeling and Its Applications to Object-Oriented Analysis, Design, and Testing*, Proc. of IEEE WORDS 2002 (IEEE Computer Society Press, Los Alamitos 2002) pp. 140–151

24.11 W.T. Tsai, F. Zhu, L. Yu, R.J. Paul: *Rapid Verification of Embedded Systems Using Patterns* (COMPSAC, IEEE Computer Society Press, Los Alamitos 2003) pp. 466–471

24.12 F. Zhu: A Requirement Verification Framework for Real-Time Embedded Systems. Ph.D. Thesis (Dept. of Computer Sci. and Engineering, Univ. of Minnesota, Minneapolis 2002)

24.13 A. Cockburn: *Agile Software Development* (Addison Wesley, Reading, MA 2001)

24.14 W.T. Tsai, R. Paul, L. Yu, A. Saimi, Z. Cao: Scenario-Based Web Service Testing with Distributed Agents, IEICE Trans. Inf. Syst. **E86-D**(10), 2130–2144 (2003)

24.15 W.T. Tsai, A. Saimi, L. Yu, R. Paul: *Scenario-Based Object-Oriented Test Frameworks* (Proc. Third Int. Conf. Quality Software (QSIC03), IEEE Computer Society Press, Los Alamitos 2003) pp. 410–417

24.16 D. Beyer, A. Chlipala, T. Henzinger, R. Jhala, R. Majumdar: Generating Tests from Counterexamples, Proc. 26th Int. Conf. Software Engineering (ICSE'04), Scotland, UK May 2004 (IEEE Computer Society, Washington, DC 2004) 326–335

24.17 W.T. Tsai, X. Wei, Y. Chen, B. Xiao, R. Paul, H. Huang: *Developing and Assuring Trustworthy Web Services* (7th International Symposium on Autonomous Decentralized Systems (ISADS), Chengdu, China, IEEE Computer Society Press, Los Alamitos April 2005)

24.18 F. Zhu: A Requirement Verification Framework for Real-Time Embedded Systems. Ph.D. Thesis (Department of Computer Science and Engineering, Univ. of Minnesota, Minneapolis 2002)

24.19 R.M. Fujimoto: Parallel and Distributed Simulation. In: *Proc. 1999 Winter Simul. Conf.*, ed. by P.A. Farrington, H.B. Nembhard, D.T. Sturrock, G.W. Evans (ACM Press, New York 1999) pp. 122–131

24.20 W.T. Tsai, W. Song, R. Paul, Z. Cao, H. Huang: *Services-Oriented Dynamic Reconfiguration Framework for Dependable Distributed Computing* (COMPSAC, IEEE Computer Society Press, Los Alamitos Sep. 2004) pp. 554–559

24.21 W.T. Tsai, Y. Chen, Z. Cao, X. Bai, H. Huang, R. Paul: *Testing Web Services Using Progressive Group Testing*, Advanced Workshop on Content Computing (Lecture Notes in Computer Science 3309, Springer Berlin, Zhenjiang 2004) pp. 314–322

24.22 E. Clarke, O. Grumberg, D. Peled: *Model Checking* (MIT Press, Cambridge, Massachusetts 2002)

24.23 T.Y. Chen, M.F. Lau: *Test Cases Selection Strategies Based on Boolean Specifications*, Software Testing, Verification and Reliability, Vol. 11 (Wiley, Atrium, UK Sep. 2001) pp. 165–180

24.24 IEEE Std1516-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules (2000)

24.25 J. Davila, E. Gomez, K. Laffaille, K. Tucci, M. Uzcategui: MultiAgent Distributed Simulation with GALATEA, 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications 2005 (IEEE Computer Society Press, Los Alamitos 2005) 165–170

24.26 L.F. Wilson, D.J. Burroughs, A. Kumar: A framework for linking distributed simulations using software agents, Proc. IEEE **89**(2), 135–142 (Feb 2001)

24.27 B. Logan, G. Theodoropoulos: The distributed simulation of multi-agent systems, Proc. IEEE **89**(2), 174–185 (2001)

24.28 H.S. Sarjoughian, B.P. Zeigler, S.B. Hall: A Layered Modeling and Simulation Architecture for Agent-Based System Development, Proc. IEEE **89**(2), 201–213 (2001)

24.29 C. Pleeger: *Security in Computing*, 3rd edn. (Prentice Hall PTR, Indianapolis 2000)

24.30 N. Damianou, A. Bandara, M. Sloman, E. Lupu: *A Survey of Policy Specification Approaches*, Technical Report (Dept. of Computing, Imperial College of Sci. Technology and Medicine, London, UK 2002)

24.31 M. Kangasluoma: *Policy Specification Languages*, Technical Report (Dept. of Computer Science, Helsinki Univ. of Technology, Helsinki Nov. 1999)

24.32 N. Damianou, N. Dulay, E. Lupu, M. Sloman: *The Ponder Policy Specification Language*, Proceedings of Workshop on Policies for Distributed Systems and Networks, 2001

24.33 L. Kagal, Rei: *A Policy Language for the Me-Centric Project*, Technical Report (HP Laboratories, Palo Alto, CA 2002)

24.34 D. Bell, L. LaPadula: *Secure Computer System: Unified Exposition and Multics Interpretation*, Technical Report (MITRE Corporation, Bedford, MA March 1976)

24.35 R. Sandhu, E. Coyne, H. Feinstein, C. Youman: Role-based access control models, IEEE Comput. **29**(2), 38–47 (1996)
24.36 E. Bertino: RBAC models – concepts and trends, Comput. Security **22**(6), 511–514 (2003)
24.37 S. Osborn, R. Sandhu, Q. Nunawer: Configuring role-based access control to enforce mandatory and discretionary access control policies, ACM Trans. Inf. Syst. Security **3**(2), 85–106 (2000)
24.38 Y. Chen: Modeling Software Operational Reliability under Partition Testing. In: *IEEE 28th Annual International Symposium on Fault-Tolerant Computing (FTCS-28), Munich, June 1998* (IEEE Computer Society Press, Los Alamitos, CA 1998) pp. 314–323
24.39 Y. Chen, J. Arlat: An Input Domain-Based Software Reliability Growth Model under Partition Testing with Fault Corrections, Proc. 15th Int. Conf. Computer Safety, Reliability, Security (SAFECOMP'96), Vienna October 1996 (Springer, Berlin 1997) 136–145
24.40 W. T. Tsai, D. Zhang, Y. Chen, H. Huang, R. Paul, N. Liao: A Software Reliability Model for Web Services. In: *8th IASTED Int. Conf. Software Eng. Appl., Cambridge MA, November 2004*, ed. by M. H. Hamza (ACTA Press, Calgary, Canada 2004)
24.41 D. C. Montgomery: *Design and Analysis of Experiments*, 5th edn. (Wiley, Indianapolis 2000)
24.42 B. Sleeper: The five missing pieces of SOA. www.infoworld.com/article/04/09/10/37FEwebservmiddle_1.html (2004)
24.43 R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, W. R. Swartout: Enabling technology for knowledge sharing, AI Magazine **12**(3), 36–56 (1991)
24.44 R. Fikes, A. Farquhar: Distributed repositories of highly expressive reusable ontologies, IEEE Intell. Syst. **14**(2), 74–79 (1999)
24.45 W. T. Tsai, R. Paul, Z. Cao, L. Yu, A. Saimi, B. Xiao: Verification of web services using an enhanced UDDI server, Proc. IEEE WORDS , 131–138 (2003)
24.46 W. T. Tsai, Y. Chen, R. Paul, N. Liao, H. Huang: Cooperative and group testing in verification of dynamic composite web services. In: *Workshop on Quality Assurance and Testing of Web-Based Applications, in conjunction with COMPSAC* (IEEE Computer Society Press, Los Alamitos, CA Sep. 2004) pp. 170–173

25. Statistical Models in Software Reliability and Operations Research

Statistical models play an important role in monitoring and control of the testing phase of software development life cycle (SDLC). The first section of this chapter provides an introduction to software reliability growth modeling and management problems where optimal control is desired. It includes a brief literature survey and description of optimization problems and solution methods.

In the second section a framework has been proposed for developing general software reliability models for both testing and operational phases. Within the framework, pertinent factors such as testing effort, coverage, user growth etc. can be incorporated. A brief description of the usage models have been provided in the section. It is shown how a new product sales growth model from marketing can be used for reliability growth modeling. Proposed models have been validated on software failure data sets.

To produce reliable software, efficient management of the testing phase is essential. Three management problems viz. release time, testing effort control and resource allocation are discussed in Sects. 25.2 to 25.4. The operations research approach, i. e. with the help of the models, optimal management decisions can be made regarding the duration of the testing phase, requirement and allocation of resources, intensity of testing effort etc. These optimization problems can be of interest to both theoreticians and software test managers. This chapter discusses both of these aspects viz. model development and optimization problems.

A scientific way of solving decision-making problems arising in large and complex systems involves the construction of a model (usually a mathematical model) that represents the character of the problem. Modeling can be the most practical way of studying the behavior of such systems. A model exhibits relationships between quantitative variables under a definite set of assumptions that portray the system. It allows experimentation with different alternative courses of actions and facilitates the use of sophisticated mathematical techniques and computers for the purpose. Mathematical models have proved to be useful for understanding the structure and functioning of a system, predicting future events and prescribing the best course of actions under known constraints. The success and popularity of operations research, a problem-solving approach started with the above philosophy as its basic working principle, has demonstrated the utility of mathematical modeling. One of the fields where mathematical modeling, particularly stochastic modeling, has been applied widely is

reliability. Stochastic modeling in reliability theory has continued to be an area of extensive research for more than four decades. The subject has traditionally been attached to hardware systems. But with ever increasing use of computers in present times software reliability has also emerged as a discipline of its own. This chapter endeavors to develop new mathematical models for software reliability evaluation and propose methods for efficient management of the testing phase.

The last decade of the 20th century will be noted in history for the incredible growth in information technology. The proliferation of the Internet has gone far beyond even the most outrageously optimistic forecasts. Consequently computers and computer-based systems have invaded every sphere of human activity. As more systems are being automated mankind's dependence on computers is rapidly increasing. Though this technology revolution has made our lives better, concern for safety and security has never been greater. There are already numerous instances where the failure of computer-controlled systems has led to colossal loss of human lives and money. Computer-based systems typically consist of hardware and software. Quality hardware can now be produced at a reasonable cost but the same cannot be said about software. Software development consists of a sequence of activities where perfection is yet to be achieved. Hence there is every possibility that fault can be introduced and can remain in a software. A fault occurs when a human makes a mistake, called an error, in performing some software activity. These faults can lead to failures with catastrophic results. Therefore a lot of emphasis is put on avoiding the introduction of faults during software development and to remove dormant faults before the product is released for use.

The testing phase is an extremely important component of the software development life cycle (SDLC), where around half the developmental resources are consumed. In this phase the software product is tested to determine whether it meets the requirement. It is endeavored to remove faults lying dormant in the software. The theory developed in this chapter primarily addresses the testing phase. The only way to verify and validate the software is by testing. The software testing involves running the software and checking for unexpected behavior of the software output. A successful test can be considered to be one that reveals the presence of latent faults. During testing, resources such as manpower and time are consumed. A very specialized kind of manpower is required for test-case generation, running the test cases and debugging. Time is also a very important resource as software cannot be tested indefinitely and there is always pressure to release the software as early as possible. With the increasing importance of cost and time during software development, efficient management of the testing phase becomes a high-priority issue for an organization. Therefore it is important to understand the failure pattern and faults causing these failures. The chronology of failure occurrence and fault removal can be utilized to provide an estimate of software reliability and the level of fault content. A software reliability model is a tool that can be used to evaluate the software quantitatively, develop test status and monitor the change in reliability performance [25.1]. Numerous software reliability growth models (SRGMs), which relate the number of failures (faults identified) and execution time (CPU time/calendar time), have been discussed in the literature [25.2–9]. These models are used to predict fault content and the reliability of software. The majority of these models can be categorized as nonhomogeneous Poisson process (NHPP) models as they assume a NHPP model to describe the failure phenomenon [25.3,6,8,10]. New models exploit the mean-value function of the underlying NHPP by proposing new forms for it; this chapter takes this modeling approach. Moreover the expected behavior of the users of the software has also been included in the modeling process.

Large software systems contain several million lines of code. The sheer size of the product presents unique problems in terms of the ability of the software designers to achieve software quality rapidly. the testing phase, which consumes the largest portion of software development resources, poses formidable challenges. Manpower from diverse background are involved in the testing process. It is this phase where a closer interaction with the users is a must. It is a fact that if software is tested for a longer period it would result in an increase in reliability. But the cost of testing also increases. Very often test managers work under tight schedules and with limited resources. Therefore, to produce reliable software, efficient management of the testing phase is required. Three such management problems viz. the release-time problem, the testing effort control problem and the resource allocation problem are discussed in this chapter.

To know when to stop testing is a pertinent question during the testing phase. If the time of release of the software for operation can be forecasted beforehand it can help management immensely. The predictive ability of SRGMs can provide a scientific answer. The release time should be an optimal tradeoff between cost and reliability. Due to the obvious importance of the problem it has received a lot of attention from re-

searchers [25.3, 8, 11–16]. In this chapter this problem has been chosen as the first illustration of the application of the methods of operations research in software reliability engineering. The usage-based SRGMs have been applied in more realistic mathematical programming formulations of the problem. Often the target reliability level is fixed for release time during the testing of software. Using SRGMs the reliability of the software can be forecasted for any future time. If it is found that the target cannot be achieved, the testing effort needs to be accelerated. The additional resource requirements can be calculated using

SRGMs [25.17–19]. In Sect. 25.3 we discuss the above testing effort control problem and provide a new solution method through an SRGM specially developed for the purpose. Optimal resource allocation is a problem that bothers all decision makers. Hence the literature in this area is very rich. Many operations researchers are working on new allocation problems arising in systems that are changing due to the proliferation of technology. Module testing in the testing phase is one such activity where optimal resource allocation can be important for obtaining reliable software [25.17, 20, 21]. In this chapter the mathematical programming approach has been suggested for the solution of this problem.

The objective of this chapter is to highlight the importance of modeling and optimization in software reliability engineering. The first part is devoted towards model development and, thereafter, three illustrations of optimization and control are provided.

25.1 Interdisciplinary Software Reliability Modeling

A commercial software developer endeavors to make its software product popular in the market by providing value to its customer and thus generating goodwill. Apart from satisfying customers by meeting all their requirements and attaching additional features, the developer at the same time makes constant efforts to build bug-free software. As manual systems are increasingly being automated, a failure due to software can lead to loss of money, goodwill and even human lives. The competition in the commercial software market is intense and, because of the nature of the applications involved, purchasers look for quality in terms of the reliability of the software. Therefore software developers lay special emphasis on testing their software.

During the testing phase, test cases that simulate the user environment are run on the software and any departure from the specifications or requirements is called a failure and an effort is made immediately to remove the cause of that failure (a fault in the software). Testing goes on until the management is satisfied with the reliability of the software. But software cannot be tested exhaustively within a limited time period. This is the reason why we often hear about failures of software in operation and sometimes even in safety-critical systems. These failures are caused by faults that remain even after testing. Hence it is important to study how these failures occur in the user phase. Selling a software is not a one-off deal. It involves cultivating long-term relationship with the purchasers. Many of the developers come up with newer versions of their software after the launch of their product. These new versions can contain codes of the previous version with some additional modules and modifications. Moreover, some developers give warranties on their products. Hence any fault that is reported by the user is corrected. If the number of faults remaining in the software can be estimated to a reasonable accuracy it can give the management a useful metric to be used for decision making under the situations discussed above. Mathematical modeling can help in developing such a metric.

SRGMs have been widely used to estimate the reliability of software during testing. Many authors have even tried to extend them to represent the failure phenomenon during the operational phase, typically used in the software release-time problem [25.3, 15, 16]. But this approach is not correct when usage of software is different from during testing, which is actually the case for most commercial software packages. Testing is done under a controlled environment. Testing resources such as manpower and consumed (computer) time can be measured and extended further into the future. Mathematical models have been proposed for testing effort itself but they are not suitable for measuring the usage of software in the market. The intensity with which failures would manifest themselves during operational use is dependent upon the number of times the software is used and not much has been done in the literature for this situation [25.21]. An attempt has been made in this chapter to model reliability growth, linking it to the number of users in the operational phase. In this chapter we propose a framework for model development for the operational phase, which can also connect the testing phase, thus providing a unique approach to modeling both the testing and operational phases.

Kenny [25.21] proposed a model to estimate the number of faults remaining in the software during its operational use. He has assumed a power function to represent the usage rate of the software. Though Kenny argues that the rate at which the commercial software is used is dependent upon the number of users, the model proposed by him can fail to capture the growth in the number of users of a software product. A mathematical model to capture the growth of users is integrated into the proposed software reliability growth model.

Although commercial software products have been on the market for two decades, identifying the target customers with certainty is impossible. Hence a product, which may be similar in many respect to another one when launched in the market, behaves as a new product or innovation. The Bass model for innovation diffusion [25.22] in marketing has satisfactorily been used for this dynamic market of software products [25.23]. This model explicitly categorizes the customers into innovators and imitators. Innovators have independent decision-making abilities whereas imitators make the purchase decisions after getting first-hand opinion from a user. Here it is assumed that purchasers or users whose number with respect to time can be modeled as an innovation diffusion phenomenon are those who can report a failure caused by the software to the developer. Such a model can correctly describe the growth of users in terms of:

1. a slow start but a gain in growth rate,
2. a constant addition of users,
3. a big beginning and tail off in the usage rate, as pointed out by *Kenny* [25.21].

The model can also describe the situation where a much-hyped product when launched in a market does not fare according to expectation. Once the number of users of the software is known, the rate at which instructions in the software are executed can be estimated. The intensity with which failures would be reported depends upon this usage. The models developed in the software reliability engineering literature can now be used to model the fault exposure phenomenon.

Another important factor that affects software reliability immensely is testing coverage, but very few attempts have been made in the literature to include its impact [25.24, 25]. With the running of test cases and corresponding failure-removal processes during the testing phase, more portions of the software, paths, functions are tested. However, it is also a fact that software cannot be tested exhaustively. As testing coverage increases software becomes more reliable. Hence testing coverage is very important for both software test managers as well as users of the software. The model developed in this chapter for both the testing and operational phases also takes this factor into account, which is another novel feature of the chapter.

Notations:

$m, m(t)$: Expected number of faults identified in the time interval $(0,t]$ during the testing phase.

$\hat{m}, \hat{m}(t)$: Expected number of faults identified in the time interval $(0,t]$ during the operational phase.

$e, e(t)$: Expected number of instructions executed on the software in the time interval $(0,t]$.

$W, W(t)$: Cumulative testing effort in the time interval $(0,t]$; $\frac{\mathrm{d}}{\mathrm{d}t}W(t) = w(t)$.

a: Constant representing the number of faults lying dormant in the software at the beginning of testing.

$p, p(W(t))$: Testing coverage as a function of time testing effort.

$\alpha, \beta, \delta, \gamma$: Constants.

g, h, k, i, j: Constants.

$k_i, i = 1, \ldots, 9$: Constants.

$\bar{W}$: Constant representing the saturation point for the growth of users of the software.

T: Release time of the software.

$m^*(t)$: Number of failures reported during the operational phase, $t > T$.

q: Factor by which the operational usage rate differs from the testing rate per remaining faults.

$R_{\mathrm{te}}(x|t)$: Reliability of the software during the testing phase, $t < T$.

$R_{\mathrm{op}}(x|t)$: Reliability of the software during the operational phase, $t > T$.

C_1, C_2: Costs of testing per unit time and removing a fault, respectively, during the testing phase.

C_3: Cost of a failure and removing it during the operational phase.

T_{s}: Scheduled delivery time of the software.

$p_{\mathrm{c}}(t)$: Penalty cost $\begin{cases} 0, & t \le T_{\mathrm{s}} \\ p_{\mathrm{c}}.t & \text{otherwise} \end{cases}$.

T_{w}: Warranty period of the software.

25.1.1 Framework for Modeling

As discussed in the Introduction, several quantitative measures of growth in reliability of software during the testing phase have been proposed in the literature and several of these can be classified as NHPP models [25.3, 8, 9]. These NHPP models are based on the assumption that 'Software failures occur at random times during testing caused by faults lying dormant in the software'. The assumption appears true for both the testing and operational phases. Hence NHPP models can be used to describe the failure phenomenon during both of these phases. The counting process $\{N(t), t \geq 0\}$ of an NHPP is given as follows.

$$\Pr[N(t) = k] = \frac{[m(t)]^k}{k!} \mathrm{e}^{-m(t)}, \quad k = 0, 1, 2,$$
$$\text{and } m(t) = \int_0^t \lambda(x)\,\mathrm{d}x . \tag{25.1}$$

The intensity function $\lambda(x)$ (or the mean-value function $m(t)$) is the basic building block of all the NHPP models existing in the software reliability engineering literature. These models assume diverse testing environments such as the distinction between failure and removal processes, learning of the testing personnel, the possibility of imperfect debugging and error generation etc. In models proposed by *Yamada* et al. [25.26] and *Trachtenberg* [25.27], the effect of the intensity of testing effort on the failure phenomenon has been studied. Faults if present in the software are exposed when the software is run. During the testing phase, test cases are run and in the operational phase the software is used by the user. Hence the rate at which failures would occur depends upon its usage (i. e. testing effort during testing or number of users in the operational phase [25.21]). Hence SRGMs should incorporate the effect of usage. But this may give rise to more complication and confusion as a number of functions exist in the literature that describes the testing effort or user growth with time. In this chapter an attempt has been made to address this problem. A general framework for model development has been proposed here. Using the basic building blocks of this framework SRGMs for both testing and operational phases can be developed with ease. The proposed approach is based upon the following basic assumptions.

1. Software failure phenomenon can be described by an NHPP. Software reliability growth models of this chapter are the mean-value functions of NHPP.
2. The number of failures during testing/operation is dependent upon the number of faults remaining in the software at that time. It is also dependent upon the rate of testing coverage.
3. Testing coverage increases due to testing effort.
4. As soon as a failure occurs the fault causing that failure is immediately identified. Identified faults are removed perfectly and no additional faults are introduced during the process.
5. The number of instructions executed is a function of testing effort/number of users.
6. Testing effort/number of users is a function of time.

Using the above assumptions the failure phenomenon can be described with respect to time as follows [25.21, 27]:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{\mathrm{d}m}{\mathrm{d}e}\frac{\mathrm{d}e}{\mathrm{d}W}\frac{\mathrm{d}W}{\mathrm{d}t} . \tag{25.2}$$

We discuss below individually each component (fraction) on the right-hand side of the above expression.

Component 1

During testing instructions are executed on the software and the output is matched with the expected results. If there is any discrepancy a failure is said to have occurred. Effort is made to identify and later remove the cause of the failure. The rate at which failures occur depends upon the number of faults remaining in the software [25.10]. As the coverage of the software is increased more faults are removed. The rate at which additional faults are identified is directly dependent upon the rate at which software is covered through additional test cases being run [25.24, 25]. It is also dependent upon the size of the uncovered portion of the software. Based on these facts the differential equation for fault identification/removal can be written as:

$$\frac{\mathrm{d}m}{\mathrm{d}e} = k_1 \frac{p'}{c-p}(a-m) , \tag{25.3}$$

where p' is the rate (with respect to testing effort) with which the software is covered through testing, c is the proportion of total software which will be eventually covered during the testing phase, with $0 < c < 1$. If c is closer to 1, one can conclude that test cases were efficiently chosen to cover the operational profile. For a logistic fault removal rate we can assume the following

form for $\frac{p'}{c-p}$:

$$\frac{p'}{c-p} = \frac{g}{1+h\mathrm{e}^{-gW}} \,. \tag{25.4}$$

$$\text{Hence, } p(W) = c\frac{1-\mathrm{e}^{gW}}{1+h\mathrm{e}^{-gW}} \,. \tag{25.5}$$

Testing coverage is directly related to testing effort, because with more testing effort we can expect to cover a larger portion of the software. Testing effort can be modeled as a function of time, which will be discussed later in this chapter.

Component 2

The second component of expression (25.2) relates the number of instructions executed with the testing effort or the number of users of the software. For the sake of simplicity we assume it to be constant

$$\frac{\mathrm{d}e}{\mathrm{d}W} = k_2 \,. \tag{25.6}$$

Substituting (25.3) and (25.6) into (25.2) we have

$$\frac{\mathrm{d}m}{\mathrm{d}t} = k_1\frac{g}{1+h\mathrm{e}^{-gW}}(a-m)k_2\frac{\mathrm{d}W}{\mathrm{d}t} \,. \tag{25.7}$$

In the next section the mathematical models for the software testing effort (component 3 of (25.2)) are discussed.

25.1.2 Modeling Testing Effort

The resources that govern the pace of testing for almost all software projects [25.6] are

1. Manpower, which includes
   - Failure-identification personnel,
   - Failure-correction personnel.
2. Computer time

In the literature, either the exponential or Rayleigh function has been used to explain the testing effort. Both can be derived from the assumption that, the testing effort rate is proportional to the testing resources available.

$$\frac{\mathrm{d}W(t)}{\mathrm{d}t} = c(t)[\alpha - W(t)] \,, \tag{25.8}$$

where $c(t)$ is the time-dependent rate at which testing resources are consumed with respect to remaining available resources. Solving (25.8) under the initial condition $W(0) = 0$, we get

$$W(t) = \alpha\left\{1 - \exp\left[\int_0^t c(x)\,\mathrm{d}x\right]\right\} \,. \tag{25.9}$$

When $c(t) = c$, a constant

$$W(t) = \alpha\left(1 - \mathrm{e}^{-ct}\right) \,. \tag{25.10}$$

If $c(t) = ct$, (25.8) gives a Rayleigh-type curve

$$W(t) = \alpha\left(1 - \mathrm{e}^{-ct^2/2}\right) \,. \tag{25.11}$$

Huang et al. [25.28] developed an SRGM, based upon an NHPP with a logistic testing-effort function. The cumulative testing effort consumed in the interval $(0, t]$ has the following form

$$W(t) = \frac{p}{1+r\mathrm{e}^{-lt}} \,. \tag{25.12}$$

Where p, r and l are constants. SRGMs with logistic testing-effort functions provide better results on some failure data sets.

Yamada et al. [25.29] described the time-dependent behavior of testing-effort expenditure by a Weibull curve while proposing an SRGM of

$$W(t) = \alpha\left(1 - \mathrm{e}^{-\beta t^k}\right) \,. \tag{25.13}$$

Exponential and Rayleigh curves become special cases of the Weibull curve for $k = 1$ and $k = 2$ respectively. To study the testing-effort process, one of the above functions can be chosen. In the following section we develop an SRGM where the fault-detection rate is a function of the testing effort and can have one of the forms discussed above.

25.1.3 Software Reliability Growth Modeling

Any one of the testing-effort models can be substituted in (25.7) to obtain a general software reliability growth model. Equation (25.7) can be written as follows

$$\frac{(\mathrm{d}m/\mathrm{d}t)}{(\mathrm{d}W/\mathrm{d}t)} = k\frac{g}{1+h\mathrm{e}^{-gW}}(a-m) \,, \tag{25.14}$$

where, $k = k_1k_2$. Equation (25.14) is a first-order linear differential equation. Solving it with the initial conditions $m(0) = 0$ and $W(0) = 0$ we have,

$$m[W(t)] = a\frac{\left(1+h\mathrm{e}^{-gW(t)}\right)^k - (1+h)\mathrm{e}^{-gkW(t)}}{\left(1+h\mathrm{e}^{-gW(t)}\right)^k} \,. \tag{25.15}$$

Next it is shown how a similar modeling approach can be used to obtain a failure-count model for the operational phase.

This SRGM is flexible and general in nature. For different parameter values it can reduce to many well-known SRGMs. *Pham* et al. [25.30] have proposed an alternative approach for the development of a general, flexible SRGM, though the impact of testing coverage was not explicitly considered in their model development. Moreover the modeling approach of this chapter can be extended to the operational phase, as shown next.

25.1.4 Modeling the Number of Users in the Operational Phase

During the operational phase failures are reported by the users. Software developers remove faults that cause these failures in future releases of the software. The number of failure reports can depend on the number of users of the software. As the usage grows so does the number of failure reports. Hence usage during the operational phase plays a similar role as testing effort during the testing phase. The failure-count model for the operational phase is based upon the following assumptions:

1. The number of unique failure reports and corresponding fault removals of the software during the operational phase can be described by an NHPP.
2. The number of failures during operation is dependent upon the number of faults remaining in the software. It is also directly proportional to the size of the uncovered portion (at the completion of the testing phase) of the software and the volume of instructions executed.
3. Once a failure is reported, the same failure report by other users is not counted. The SRGM developed can be interpreted as a failure-count model. The debugging process by the developer is assumed to be perfect.
4. The volume of instructions executed is related to the number of users.
5. The number of users of the software is a function of time.

Using the above assumptions the fault-removal phenomenon during the operational phase can be described as a function of time as follows:

$$\frac{\mathrm{d}\hat{m}}{\mathrm{d}t} = [1 - p(T)]\frac{\mathrm{d}\hat{m}}{\mathrm{d}e}\frac{\mathrm{d}e}{\mathrm{d}W}\frac{\mathrm{d}W}{\mathrm{d}t}\,, \tag{25.16}$$

where T is the release time of the software, $[1 - p(T)]$ is the size of the uncovered portion of the software and its value is known at the time of release of the software, $\hat{m}$ is the mean-value function of the failure-count model for the operational phase, i. e., the expected number of faults removed during the operational phase. The other three fractions of the right-hand side of (25.16) can be modeled similarly to the process followed in Sect. 25.1.1. Now fault removal is directly dependent on the number of instructions executed. It is also a fact that additional faults are removed during code checking for failure-cause isolation, but these faults may not have caused failures. *Kapur* and *Garg* [25.31] have discussed this phenomenon. Based upon these arguments the following expression can be written

$$\frac{\mathrm{d}\hat{m}}{\mathrm{d}e} = \left(k_4 + k_5\frac{\hat{m}}{a_1}\right)(a_1 - \hat{m})\,, \tag{25.17}$$

where k_4 is the rate at which remaining faults cause failures. It is a constant, as each one of these faults has an equal probability of causing failure. k_5 is the rate at which additional faults are identified without causing any failures; it is a constant, but also depends upon the number of faults already identified. $a_1 = [a - m(T)]$ is the number of faults present in the software when it was released for use (test time T).

During the debugging process some of the faults might be imperfectly removed and can cause failure in future. If this factor is introduced into the model, (25.17) can be modified as follows:

$$\frac{\mathrm{d}\hat{m}}{\mathrm{d}e} = \left(k_4 + k_5\frac{\hat{m}}{(a_1 + \hat{m})}\right)\left[(a_1 + k_6\hat{m}) - \hat{m}\right]. \tag{25.18}$$

In the above expression k_6 is the rate of imperfect debugging. But finding a closed-form solution for (25.18) is difficult. Therefore we can assume a logistic rate function as discussed (25.17),

$$\frac{\mathrm{d}\hat{m}}{\mathrm{d}e} = \left(\frac{i}{1 + j\,\mathrm{e}^{-it}}\right)(a_1 - k_7\hat{m})\,, \tag{25.19}$$

where, $k_7 = 1 - k_6$. Moreover, assuming that the number of instructions executed is a constant with respect to usage growth, the following expression, which is similar to (25.6), can be written.

$$\frac{\mathrm{d}e}{\mathrm{d}W} = k_8\,. \tag{25.20}$$

25.1.5 Modeling the User Growth

Kenny [25.21] used the power function to describe the growth in the user population of a software

$$W(t) = \frac{t^{(k+1)}}{(k+1)} \,. \tag{25.21}$$

$W(t)$ here is the number of users of the software in the operational phase at time t. The function can correctly describe the users growth in terms of

1. a slow start but a gain in growth rate,
2. a constant addition of users,
3. a big beginning and tail off in the usage rate.

However, in the marketing literature, the power function is seldom used for the purpose as described above. One of the reasons may be that the parameters of the function are not amenable to interpretation. The growth in the number of users with respect to time can also be described by the Bass model [25.22] of innovation diffusion. To apply the Bass model it is assumed that there exists a finite population of prospective users who, with time, increasingly become actual users of the software (no distinction is made between users and purchasers here as the Bass model has been successfully applied to describe the growth in number of both of them). In each period there will be both innovators and imitators using the software product. The innovators are not influenced in their timing of purchase by the number of people who have already bought it, but they may be influenced by the steady flow of nonpersonal promotion. As the process continues, the relative number of innovators will diminish monotonically with time. Imitators are, however, influenced by the number of previous buyers and increase relative to the number of innovators as the process continues.

The combined rate of first purchasing of innovators and imitators are given by the term $\left(\alpha + \beta \frac{W(t)}{\bar{W}}\right)$ and increases through time because $W(t)$ increases through time. In fact the rate of first purchasing is shown as a linear function of the cumulative number of previous first purchasers. However, the number of remaining nonadopters, given by $\left[\bar{W} - W(t)\right]$ decreases through time. The shape of the resulting sales curve of new adopters will depend upon the relative rate of these two tendencies. If a software product is successful, the coefficient of imitation is likely to exceed the coefficient of innovation i. e. $\alpha < \beta$. On the other hand, if $\alpha > \beta$, the sales curve will fall continuously.

The following mathematical model, known as the Bass model [25.22] in the marketing literature, describes this situation.

$$\frac{\mathrm{d}W(t)}{\mathrm{d}t} = \left(\alpha + \beta \frac{W(t)}{\bar{S}}\right)\left[\bar{S} - W(t)\right] . \tag{25.22}$$

The solution of (25.22) for $W(t=0)$ is

$$W(t) = \bar{W}\frac{1 - \exp[-(\alpha+\beta)t]}{1 + (\beta/\alpha)\exp[-(\alpha+\beta)t]} \,. \tag{25.23}$$

Givon et al. [25.23] have used the modified version of this model to estimate the number of licensed users as well as users of pirated copies of the software. Though it can be reasonably assumed that it is the licensed-copy holders who would report the failures, (25.23) can be used to find the expected number of users at any time during the life cycle of the software. If the new software is expected to go through the same history as some previous software (very likely for versions of the same software) the parameters of an earlier growth curve may be used as an approximation.

The derivative of (25.23) to be used in expression (25.16) has the following form

$$\frac{\mathrm{d}W(t)}{\mathrm{d}t} = \bar{W}\frac{\beta[1 + (\beta/\alpha)]\exp[-(\alpha+\beta)t]}{\left\{1 + (\beta/\alpha)\exp[-(\alpha+\beta)t]\right\}^2} \,. \tag{25.24}$$

After substitution of all the components, (25.16) is a first-order linear differential equation. Solving it with the initial condition $m(t=0) = 0$ we have,

$$\hat{m}(t) = \frac{a_2}{k_9}\left[1 - \left(\frac{(1+j)\mathrm{e}^{-\mathrm{i}W(t)}}{1 + j\,\mathrm{e}^{-\mathrm{i}W(t)}}\right)^{k_9}\right] , \tag{25.25}$$

where $a_2 = k_8 a_1[1 - p(T)]$ and $k_9 = k_7 k_8[1 - p(T)]$.

25.1.6 Estimation Methods

The testing-effort data or the data pertaining to the number of users (or usage) of a software can be collected in the form of testing effort/usage, $w_k(- < w_1 < w_2 < \ldots < w_n)$ consumed in time $(0, t_i]$; $i = 1, 2, \ldots, n$. Then the testing-effort model/usage growth model parameters can be estimated by the method of least squares as follows

$$\text{Minimize} \sum_{i=1}^{n} \left(W_i - \hat{W}\right)^2 \tag{25.26}$$

subject to $\hat{W}_n = W_n$ (i. e. the estimated value of the testing effort is equal to the actual value).

To estimate the parameters of the SRGMs obtained through (25.15) and (25.25), the method of maximum likelihood (MLE) is used [25.3, 6, 8]. The fault-removal

Table 25.1 Fitting of testing effort data

Data sets	α	β	k	R^2
DS-1	2669.9	0.000773	2.068	0.99
DS-2	11710.7	0.0235	1.460154	0.98

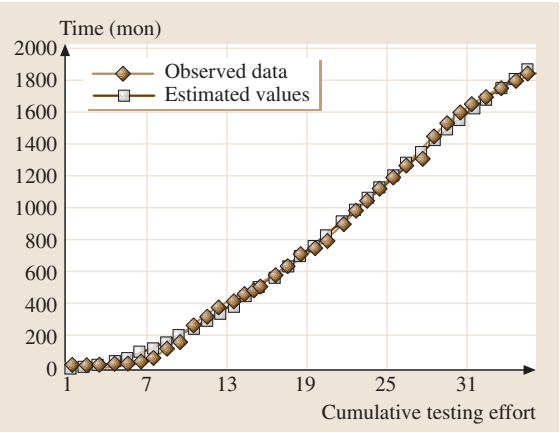


Fig. 25.1 Fitting of the effort curve (DS-1)

data is given in the form of cumulative number of faults removed, y_i in time $(0, t_i]$. Thus the likelihood function is given as

$$L[a_1, a_2, b_o, b_r, q|(y_i, W_i)] = \prod_{i=1}^{n} \frac{[m(t_i) - m(t_{i-1})]^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} e^{-[m(t_i) - m(t_{i-1})]} . \quad (25.27)$$

25.1.7 Numerical Illustrations

To validate the models four real software failure data sets have been chosen. The first two were collected during the testing phase of software while the third and fourth data sets are based on failure reports of software in operational use.

Data set 1 (DS-1): The data are cited from *Brooks* and *Motley* [25.11]. The fault data set is for a radar system of size 124 KLOC (kilo lines of code) tested for 35 months, in which 1301 faults were identified.

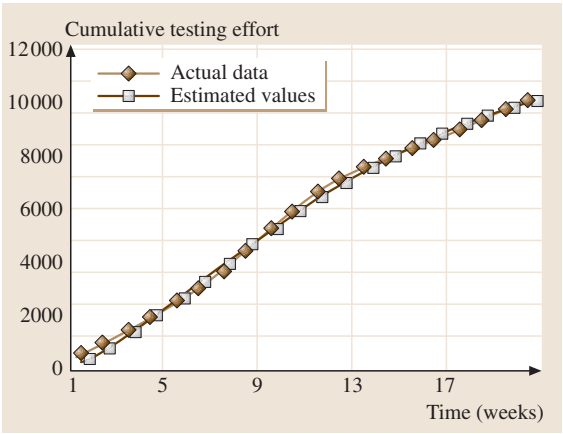


Fig. 25.2 Fitting of the testing effort curve (DS-2)

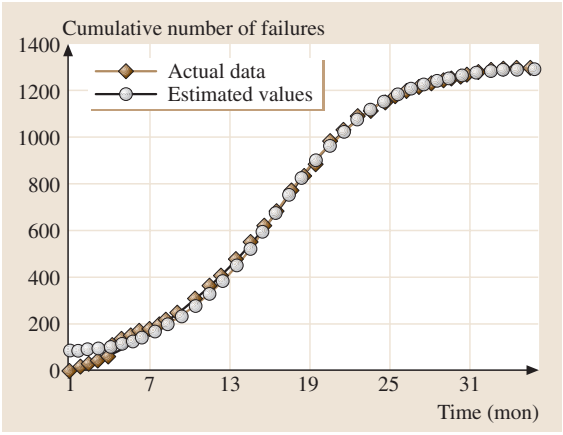


Fig. 25.3 Fitting of the failure curve (DS-1)

Data set 2 (DS-2): The data set pertains to release 1 of the tandem computer project cited in [25.30]. The software test data is available for 20 weeks, during which 100 faults were identified.

In both cases the Weibull function (25.13) gave the best fit to the testing-effort data. The results are presented in Table 25.1 and the curve fits are depicted in Figs. 25.1 and 25.2, respectively.

The estimation results for the parameters of SRGM (25.15) have been summarized in Table 25.2 and are graphically presented in Figs. 25.3 and 25.4.

Table 25.2 Parameter estimation of the SRGM

Data set	a	H	g	k	R^2
DS-I	1305	4.46445	0.003173	1.0363	0.996
DS-II	110	4.8707	0.000392	1.092	0.997

Table 25.3 Estimation result on DS-3

a_2	K_9	j	i	$\overline{W}$	$(\alpha+\beta)$	β/α	R^2
112	0.978	4.352	0.026×10^{-5}	31038400	0.010634	2.4469	0.989

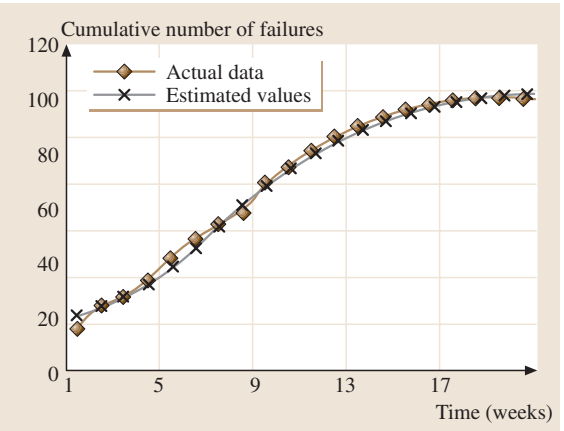


Fig. 25.4 Fitting of the failure curve (DS-2)

Next we estimate the parameters of SRGMs obtained from equation (25.25) and using usage growth functions (25.21) and (25.23). The following data set has been chosen for illustration.

Data set 3 (DS-3): This failure data set [25.32] is for an operating system in its operational phase. The software consists of hundreds of thousands of delivered object code instructions. 112 faults were reported during the observation period of around five months. The Bass model (25.23) could best describe the usage data and hence was chosen. Using the estimated values, the rest of the parameters of the model were estimated. The estimation results are summarized in Table 25.3 and are depicted in Figs. 25.5 and 25.6.

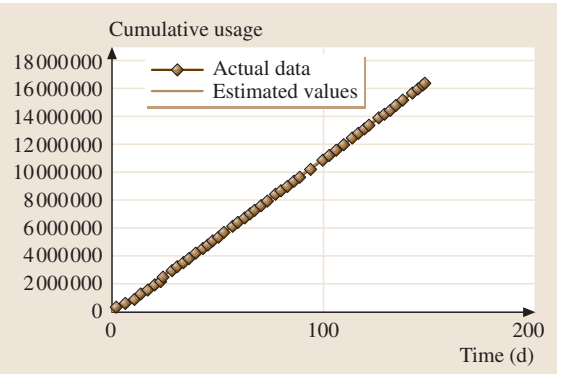


Fig. 25.5 Fitting usage data (DS-3)

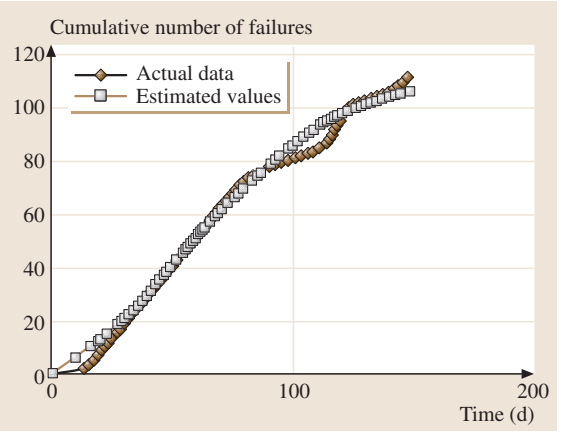


Fig. 25.6 Fitting of number of failures (DS-3)

25.2 Release Time of Software

It is important to know when to stop testing. The optimal testing time is a function of many variables: software size, level of reliability desired, personnel availability, market conditions, penalty cost due to delay in delivery of the product and penalties of in-process failures. If the release of the software is unduly delayed, the software developer may suffer in terms of penalties and revenue loss, while premature release may cost heavily in terms of fault removals to be done after release and may even harm the manufacturer's reputation. Software release-time problems have been classified in different ways. One of them is to find the release time such that the cost incurred during the remaining phases of the life cycle (consisting of the testing and operational phases) of the software is minimized [25.3, 15]. This problem can also be alternatively defined in terms of maximizing gain, where gain is defined as the difference in cost incurred when all faults are removed during the operational phase as against the cost when some faults are removed during the testing phase and others are removed during the operational phase. It can be proved that maximizing gain is the same as minimizing cost. Some

release-time problems are based upon reliability criteria alone. Models that minimize the number of remaining faults in the software or the failure intensity fall under this category [25.3]. Release-time problems have also been formulated for minimizing cost with minimum reliability requirements or maximizing reliability subject to budgetary constraints [25.3]. The bicriterion release policy simultaneously maximizes reliability and minimizes cost subject to reliability and resource constraints. In all these formulations software reliability growth models play a very important role due to their predictive ability. It is a fact that the longer software is tested, the higher its reliability. But it cannot be tested indefinitely, due to time and cost factors. With increasing cost, there is also a loss of opportunity in earning profit. Again software can have scheduled delivery time and the developer may have to pay high penalty costs due to a delay in delivery. Hence an optimal tradeoff between cost and reliability is required to find the termination time of testing. All the costs mentioned above are minimized subject to some constraints. These constraints are primarily related to a certain minimum level of testing reliability. The cost and reliability functions are discussed in detail later in this section.

The Cost Function

Cost functions discussed in the literature include costs of testing, removing faults during testing and that of failures and removals during the operational phase. As testing is done under a controlled environment, costs pertaining to testing, removing faults, documentation etc. can be estimated, but difficulty arises in quantifying the cost of a failure at the user end. As a way out a more realistic approach of warranty cost is being considered [25.33]. In release-time problems, costs of failure and removal of a fault occurring during a limited warranty period immediately after release need also to be included. Failure during the operational phase also amounts to loss of goodwill for the developer. Hence, in the cost function, failures after testing are counted and costs for their removal are estimated. Cost due to delay in delivery [25.34] is normally included in the overall software development cost. Release-time problems should include their affect. The cost function can takes the following form:

$$C(T) = c_1 T + c_2 m(T) + c_3 m^*(T + T_w) + p_c(T) \,. \tag{25.28}$$

It is assumed above that the costs of removing faults during testing and operation are constants: c_2 and c_3, respectively. The functional forms for $m(T)$ and $m^*(T + T_w)$ are also required. SRGMs have been used for $m(T)$ and the model that best describes the reliability growth during the testing phase needs to be chosen. To estimate the number of failures in the warranty period, models for the operational phase should be used. A typical cost function with an S-shaped reliability growth curve can take the form (or a part of it) of the curve, as shown in Fig. 25.7.

In the release-time problems discussed in the literature, it has been assumed that failures will occur in the operational phase in the same manner as they do during testing [25.15]. Though the testing environment is designed such that it best represents the operational phase, the intensity of use of the software may differ. It is shown below how the simple Goel–Okumoto model [25.10] can be modified for the purpose. For the failure phenomenon of software in operational phase, the following differential equation is proposed:

$$\frac{\mathrm{d}}{\mathrm{d}t} m^*(t) = b_1 [a_1 - m^*(t)] \,, \quad t > T \,. \tag{25.29}$$

This equation is based on the assumption that failures during the operational phase are dependent on the number of faults remaining in the software at and after the time of release. Again the rate at which failures will occur with respect to the remaining faults is dependent on perceived usage of the software during this period. It is also assumed that upon a failure the corresponding fault is to be removed, at least during the warranty period. Our primary interest during this phase is to count the failures, as this directly translates to very high costs on account of risk, loss of goodwill and removal of faults or replacement of the entire software. The number of faults remaining in the software at time T is,

$$a_1 = [a - m(T)] = a - m(T) \,. \tag{25.30}$$

It is expected that there would be no fault generation during debugging in this phase. It is also assumed that the

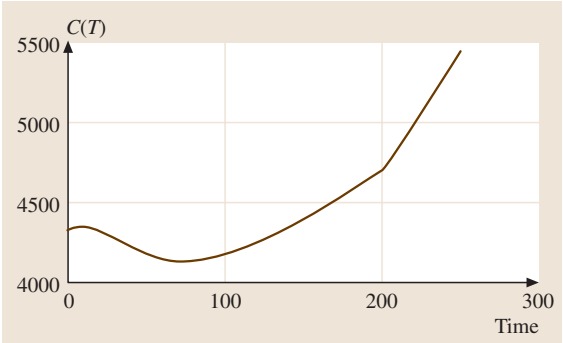


Fig. 25.7 The cost function

usage in the operational phase differs from that during the testing phase by a constant factor q. Hence the new rate is $b_1 = bq$. If $q = 1$, the intensity of use of the software during both these phases is similar. For $q < 1$ ($q > 1$) the software is expected to be used less (more) intensely during the operational phase. The solution of the differential equation (25.29) with the initial condition $m^*(T) = 0$ is

$$m^*(t) = a_1 \left[1 - e^{-b_1(t-T)}\right], \quad t > T, \qquad (25.31)$$

where $m^*(t)$ represents the expected number of failures in operational phase by time 't'. It is assumed that the failure phenomenon is still governed by an NHPP but with a new mean-value function.

Reliability Functions

Reliability expressions for NHPP software reliability models can easily be derived [25.6, 8, 10]. Software reliability is defined as the probability that the software operates failure-free for a specified time interval, on the machines for which it was designed, with the condition that the last failure occurred at a known time epoch. If the fault-detection process follows an NHPP then it can be shown that the software reliability at time t for a given interval $(t, t+x)$ is given by,

$$R_{te}(x|t) = e^{-[m(t+x)-m(t)]}. \qquad (25.32)$$

Software reliability at time 't' during the user phase is defined as the probability of nonoccurrence of failure in the interval $(t, t+x]$, $x \geq 0$, $t > T$; in the operational environment. The definition is similar to the definition for the testing phase. A mathematical expression for the same can be derived using the SRGM (25.31) and the NHPP assumption. The following expressions results [25.6, 22].

$$R_{op}(x|t) = e^{-[m^*(t+x)-m^*(t)]}, \quad t > T. \qquad (25.33)$$

The reliability curves for different values of q for a particular data set are given in Fig. 25.8. It is also observed that, for particular values of a, b and p, the operational reliability curve lies above the testing reliability curve i. e. $R_{op}(x|t) \geq R_{te}(x|t)$, $t \in [0, \infty)$ when $q = 1$. This result agrees with that derived in [25.16] when testing and operational profiles are identical.

25.2.1 Release-Time Problem Formulations

The release-time problem of software is to find a testing termination time T^* from an optimal tradeoff between cost and reliability. Many optimization problems

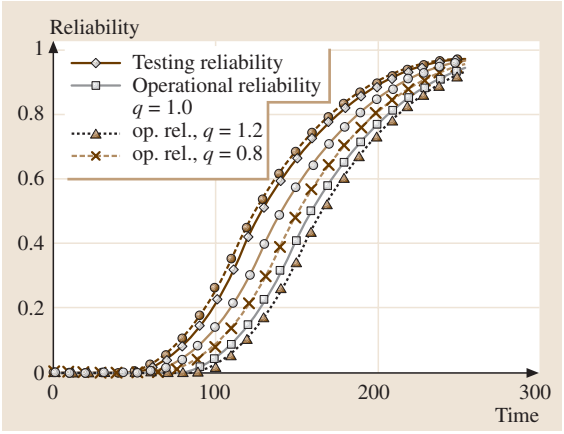


Fig. 25.8 Reliability curves for various q

have been formulated in the literature for this purpose [25.13, 15, 16, 33–36]. These problems select one or more functions from the lists of objective functions and constraints.

Objectives

(O1) Cost function: minimize $C(T)$;
(O2) Reliability functions:

1. Maximize $R_{te}(x|t)$,
2. Maximize $R_{op}(x|t)$.

Constraints

(C1) Budget constraint: $C(T) \leq B$;
(C2) Reliability constraints:

1. $R_{te}(x|t) \geq R_t$,
2. $R_{op}(x|t) \geq R_o$.

Where R_t and R_o are minimum reliability requirements for the testing and operational reliabilities respectively. In Table 25.4 some release-time problem formulations [25.3, 8] have been presented.

(R1) is the easiest problem formulation and is applicable for routinely developed software for which requirements are well defined. (R2) should be chosen for safety critical systems, where reliability is of utmost importance. (R4) is the most general problem formulation but is the most difficult to solve. A number of methods, including visual inspection of the cost curve, calculus, nonlinear programming, dynamic programming and neural networks etc. [25.3, 12, 15, 36], have been applied to find the optimal solution. Optimization software packages can also be used for this purpose.

Table 25.4 Release-time problems

Release time problem	Objective(s)	Constraints(s)
(R1) Cost criterion	Cost function (O1)	None
(R2) Reliability criterion	Reliability functions (O2-a) and (O2-b)	Reliability constraints (C2-a) and (C2-b)
(R3-A) Cost-reliability critera	Cost function (O1)	Reliability constraints (C2-a) and (C2-b)
(R3-B) Reliability-cost critera	Reliability functions (O2-a) and (O2-b)	Budget constraint (C1)
(R4) Bicriterion release criteria	Cost function (O1) Reliability functions (O2-a) and (O2-b)	Budget constraint (C1) Reliability constraints (C2-a) and (C2-b)

25.3 Control Problem

Before the release of software, a target reliability level is fixed. A reliable estimate of the fault content of software can also be obtained. Hence the management may desire to remove a certain percentage of it before release. But during the testing phase it is frequently realized that this may not be achievable for a number of reasons, such as inadequacy of the testing effort, inefficiency of the testing team etc. Hence, there is a need to increase the fault-removal rate. The problem of accelerating fault removal to achieve a certain reliability level or to a remove a certain percentage of total fault content of software is known as the testing-effort control problem. *Yamada* and *Ohtera* [25.19] took the software reliability growth modeling approach to solve this problem. In this section a new method to accelerate fault removal using SRGMs is proposed.

Additional Notations Used in this Section

α: Total testing resources to be eventually consumed, a constant;
$W_o(t)$: Cumulative testing effort on failure observation;
$W_r(t)$: Cumulative testing effort on fault removal;
$m_f(t)$: Number of failures observed in $(0, t]$;
b_o, b_r: Constants of proportionality, denoting rates of failure observation and fault removal, respectively;
m^*: Number of faults desired to be removed in time $(0, T_2]$;
$W(t - T_1)$: $W(t) - W(T_1)$, T_1 is the time duration;
a_1: $a - m(T_1)$;
a_2: $a - m_f(T_1)$.

25.3.1 Reliability Model for the Control Problem

The management of a software development project has time schedules for testing and release of software, but it is ignorant about the number and nature of faults lying dormant in it before the testing is actually done. SRGMs help in this regard after testing has been carried out for a certain period. The estimated parameters of the selected SRGM provide information about the number of faults remaining and the efficiency of the testing effort. Hence the expected number of faults that will be removed at any time in the future can be forecasted if the effort follows a known pattern. Frequently, management aspires to a reliability level at release that can be interpreted in terms of remaining number of faults. When the forecasted number of faults falls below the desired number, the testing effort needs to be controlled [25.18]. One obvious method (method I) is to increase the intensity of the testing effort through the employment of more manpower, computer time etc. But with limited resources available, this may not be feasible. Here it is shown how fault removal can also be accelerated by manipulating the allocation of testing resources to the two processes of failure observation and fault removal (method II). The models developed earlier in this chapter do not distinguish between fault identification and removal phenomenon. For the solution of the control problem we use the following SRGM.

The software testing phase aims to observe the failure process and remove the cause of the failure (the removal process). It is observed that different amounts of testing resources are consumed by each of these

processes. In SRGMs developed in the literature, the time-dependent behavior of the testing effort and the consequent reliability growth has been studied.

Yamada et al. [25.37], have given an SRGM incorporating the time lag between failure observation and fault removal. *Kapur* et al. [25.3] developed an S-shaped SRGM based on an NHPP to model the relationship between fault removal and testing effort. The cumulative testing effort was taken as a weighted sum of resources spent on fault observation and removal processes. We modify these SRGMs here. It is a common experience that, during early stages of testing, a large number of failures are observed, while the corresponding fault removals are lower. On the other hand, during later stages of testing, failures are harder to observe. Hence the failure-detection and the fault-removal processes should be studied distinctly.

Let $q_o(t)$ and $q_r(t)$ be the proportions of testing effort used on the failure-observation and fault-removal processes i. e. $q_o(t) = \frac{W_o(t)}{W_o(t)+W_r(t)} = \frac{W_o(t)}{W(t)}$ and $q_r(t) = \frac{W_r(t)}{W(t)}$. Then $q_o(t)+q_r(t) = 1$. If we assume $q_o(t) = q$ and $q_r(t) = (1-q)$, where q is a constant lying between 0 and 1. Then $qW(t)$ denotes the testing effort on failure observation and $(1-q)W(t)$, the effort on fault correction in the interval $(0, t]$. The NHPP-based SRGM developed below is based on the following assumptions:

1. No new faults are introduced into the software system during the testing phase.
2. The rate of fault removal to the current testing effort on removal is proportional to the number of identified faults that are yet to be removed at that instant.

The assumptions take the form of the following differential equations

$$\frac{m_f'(t)}{qw(t)} = b_o\,[a - m_f(t)]\ , \tag{25.34}$$

$$\frac{m'(t)}{(1-q)w(t)} = b_r\,[m_f(t) - m(t)]\ . \tag{25.35}$$

Solving the above system of equations with the initial conditions $m_f(t=0) = 0$ and $m(t=0) = 0$, we get

$$m(t) = a\left\{1 - \frac{1}{-b_o q + b_r(1-q)} \times \left[b_r(1-q)\mathrm{e}^{-b_o q W(t)} - b_o q\,\mathrm{e}^{-b_r(1-q)W(t)}\right]\right\}. \tag{25.36}$$

Equation (25.36) represents the cumulative number of faults removed, with respect to the testing effort consumed in the interval $(0, t]$. The time-dependent testing-effort function can have any of the forms presented in the preceding subsection. The removal function (25.36) is an S-shaped growth curve, because of the time lag between failure observation, the removal of the corresponding fault and the nature of the effort function.

For $q = 1$, the model reduces to the exponential model due to *Goel* and *Okumoto* [25.10]. In this case the process consists of a single step, i. e. faults are removed as soon as they are identified. With increasing $(1-q)$, the effort on removal increases. Hence the SRGM (25.36), captures the severity in faults present in a software. The model has been validated on actual software reliability data sets [25.17].

25.3.2 Solution Methods for the Control Problem

Method I

Suppose that software has been tested for time T_1 and it is to be released by time T_2, $T_2 > T_1$. Using the test data for the interval $(0, T_1]$ the parameters of the SRGM (25.36), can be statistically estimated. The testing effort in this interval is $W(T_1)$ and the corresponding number of faults that have been removed is $m(T_1)$. Based on the estimates of parameters, the number of faults expected to be removed by time T_2 is,

$$m(T_2) = a\left\{1 - \frac{1}{-b_o q + b_r(1-q)} \times \left[b_r(1-q)\mathrm{e}^{-b_o q W(T_2)} - b_o q\,\mathrm{e}^{-b_r(1-q)W(T_2)}\right]\right\} \tag{25.37}$$

The difference $[m(T_2) - m(T_1)]$ is the number of faults that is expected to be removed in the interval $(T_1, T_2]$. Often the management aspires to a level of reliability for the software at the time of release, which can be translated in terms of the number of faults (m^*) that it

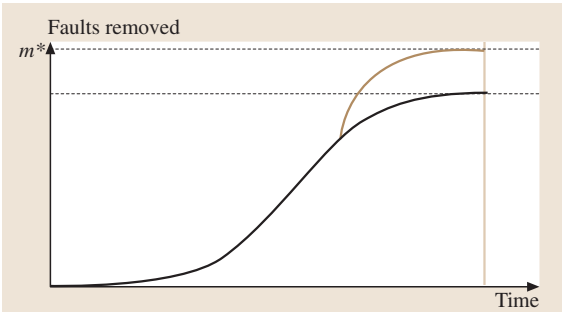


Fig. 25.9 Testing effort control

desired to be removed. If $m^* > m(T_2)$, the fault-removal rate has to be increased. This control problem is depicted in Fig. 25.9.

First the testing effort required to remove $[m^* - m(T_2)]$ faults in the time interval $(T_1, T_2]$ is calculated. Using the assumptions for the SRGM (25.36), the following expression results for $m(t)$, $t > T_2$

$$m(t) = m(T_1) + a_1\left(1 - \mathrm{e}^{-b_\mathrm{r}(1-q)W(t-T_1)}\right) - \frac{a_2 b_\mathrm{r}(1-q)}{-b_\mathrm{o}q + b_\mathrm{r}(1-q)} \times \left(\mathrm{e}^{-b_\mathrm{o}qW(t-T_1)} - \mathrm{e}^{-b_\mathrm{r}(1-q)W(t-T_1)}\right) . \tag{25.38}$$

In the above equation if m^* is substituted for $m(t)$ and W^* for $W(t-T_1)$, the following expression results,

$$m^* = m(T_1) + a_1\left\{1 - \mathrm{e}^{-b_\mathrm{r}(1-q)W^*}\right\} - \frac{a_2 b_\mathrm{r}(1-q)}{-b_\mathrm{o}q + b_\mathrm{r}(1-q)} \times \left\{\mathrm{e}^{-b_\mathrm{o}qW^*} - \mathrm{e}^{-b_\mathrm{r}(1-q)W^*}\right\} . \tag{25.39}$$

With the values of $m^*, m(T_1), a_1, b_o, a_2, b_r$ and q being known, equation (25.39) can be solved numerically to obtain the value of W^*, i. e. the amount of additional resources needed.

Method II

An alternative way to achieve the desired fault-detection level is to change the allocation factor of resources to be spent on the failure-identification and fault-removal processes. During the early stages of the testing phase a large number of failures may be observed, while the corresponding fault-removal rate is lower. This is due to the latency time needed by the removal team to cope with the workload. In this case it is reasonable to allocate resources in order to increase the testing effort of the failure-removal team, which may stimulate removal. On the other hand, during the late stages of testing, failures may be hard to identify and the removal team would have had enough time to remove most of the faults. Thus the fault removal will slow down, due to the lower number of failure observations. Hence it is more logical to assign more resources to the failure-identification team. Again during the later stages of testing it may happen that most of the failures had already been identified but not removed. Ideally the testing effort should now be concentrated on removal.

As discussed above, removal can be accelerated through proper allocation of resources. The optimal proportion of resources to be allocated to the failure-identification process, q^* to remove m^* faults can be found by solving the following equation numerically

$$m^* = m(T_1) + a_1\left(1 - \mathrm{e}^{-b_\mathrm{r}(1-q^*)W(T_2-T_1)}\right) - \frac{a_2 b_\mathrm{r}(1-q^*)}{-b_\mathrm{o}q^* + b_\mathrm{r}(1-q^*)} \times \left(\mathrm{e}^{-b_\mathrm{o}q^*W(T_2-T_1)} - \mathrm{e}^{-b_\mathrm{r}(1-q^*)W(T_2-T_1)}\right) . \tag{25.40}$$

The proportion of testing effort to be spent on fault removal during the time interval $(T_1,T_2]$ is $(1-q^*)$. In the following section, the results derived through the two methods are illustrated numerically.

In the literature only method I has been proposed for the control problem, but frequently management has to deal with limited testing resources. Method II, which increases fault removal through proper segregation of resources, can provide a solution. Moreover, following this method, testing can be done more efficiently by constantly monitoring the effort and fault removal and then allocating the optimal proportion of resources to the testing teams [25.38]. The control problem and the solution methods can be further refined. There can be an upper limit other than the initial fault content on the number of faults that can be removed by changing the allocation factor. Again the sensitivity analysis of the parameters with respect to the estimates and the optimal solution can provide further insight into the optimal allocation of testing resources. An early release of software is always desirable, but this should not undermine its quality. Also with a limited budget, decision making on the duration of testing becomes complicated. Hence cost–reliability criteria for the release of software and control of the testing effort should be jointly considered.

25.4 Allocation of Resources in Modular Software

Large software consists of modules. Modules can be visualized as independent pieces of software performing predefined tasks, mostly developed by separate teams of programmers and sometimes at different geographical locations. These modules are later integrated to form complete software. In module testing, each module is

tested independently and the software environment is simulated [25.39]. Typically this phase consumes 25% of the total development effort. In this phase the objective is to remove the maximum number of faults lying dormant in the modules. Though no conclusion can be drawn on system reliability at this stage, it is definitely enhanced with each fault removal. However, the testing has to be concluded within a specified time, which calls for proper allocation of limited resources among modules. This gives rise to the management problem of maximization of total fault removal within a finite time period or testing resource budget. In this section, we have formulated it as a mathematical programming problem. Again all modules are not equally important neither do they contain an equal number of faults. The severity of a fault in each module can also differ. In the light of this we have proposed another mathematical programming problem in that section. To arrive at a solution to this problem a mathematical relationship between testing resource consumption and fault removal is required. SRGMs have been used as a tool to monitor the progress of the testing phase by quantifying various reliability measures of the software system such as reliability growth, remaining number of faults, mean time between failures, testing effort etc. One approach is due to *Musa* et al. [25.6], where they have assumed that the resource consumption isan explicit function of the number of faults removed and calendar time. We develop a new model for the testing effort, which is solely dependent upon fault removal. The mathematical model is based upon the marginal testing effort function (MTEF), is defined as the effort required to remove an additional fault at any time. MTEFs are different in each module, depending on the severity of the faults in it, and using them we determine the optimal allocation of testing resources among modules; this has never been used before for optimal allocation of resources.

Using a simpler form of the MTEF a closed-form solution for optimal allocation of testing resource is obtained. It is very important to give a plausible form, but, as seen in another optimization problem, when more practical constraints are added to the same problem, no closed-form solution could be obtained. The problem is then solved as a nonlinear programming problem using a software package.

Additional Notations for this Section

m_i: Number of faults removed in the *i-th* module;
c_i: Cost of unit testing effort in the *i-th* module;
α_i: Relative importance of module i, $\sum_{i=1}^{N} \alpha_i = 1$;
$w(m)$: Marginal testing effort when m faults have been removed;
$W(m)$: Cumulative testing effort when m faults have been removed;
$W_i(m)$: Cumulative effort to remove m faults in the *i-th* module;
$w_i(m)$: Marginal testing effort function in the *i-th* module;
B: Total testing resource available; budget;
r_i: Minimum number of faults desired to be removed in the *i-th* module;
N: Number of modules;
k', p, q, k: Constants of proportionality;
p_i, q_i, k_i: Parameters of marginal testing effort function for the *i-th* module;
a_i: Number of faults in the *i-th* module;
d_i: $c_i k_i$;
δ, γ: Probability of imperfect (perfect) debugging, $0 \le \delta, \gamma \le 1$;
δ_i, γ_i: Probability of imperfect (perfect) debugging, $0 \le \delta_i, \gamma_i \le 1$.

25.4.1 Resource-Allocation Problem

Consider software with 'N' modules. During module testing each module is tested independently. We assume that the modules have a finite number of faults and we aspire to remove the maximum number of them. Testing resources such as manpower and computer time are used and the management has to allocate limited testing resources among the modules. This problem of optimal allocation of testing resources among modules can be formulated as a mathematical programming problem, which is given below.

$$\text{maximize} \quad \sum_{i=1}^{N} m_i \,,$$

subject to

$$\sum_{i=1}^{N} c_i W_i(m_i) = B \,. \tag{25.41}$$

Kubat and *Koch* [25.18] have used SRGMs and through the method of Lagrangian multipliers have obtained solution to the above problem. However, this method does not rule out the possibility of negative allocation of resources to some modules. To correct this, algorithms that are similar in nature have been proposed in these

papers; they sequentially give zero allocations to these modules and distribute the values among the others. But this proposition is also not suitable, as testing cannot be stopped abruptly. As a way out, management can decide upon some minimum number of faults that it expects to remove from each module.

All modules of software are not equally important. The relative importance of modules can be determined based upon the frequency with which modules are expected to be called for execution in the actual user environment. Accordingly weights can be attached to each module. Incorporating this, the optimization problem above (25.41) can be reformulated as

$$\text{maximize} \quad \sum_{i=1}^{N} \alpha_i m_i$$

subject to

$$m_i \geq r_i \,, \quad i = 1, \ldots, N \,,$$
$$\sum_{i=1}^{N} c_i W_i(m_i) = B \,. \tag{25.42}$$

A functional relationship between the testing effort and fault removal is needed before we solve (25.41) or (25.42). SRGMs can be used for this purpose. We adopt the reverse approach and develop a model for resource consumption vis-à-vis fault removal in the next section.

25.4.2 Modeling the Marginal Function

Most SRGMs depict reliability growth with reference to execution time. Only a few SRGMs incorporate the effect of a time-dependent testing-effort pattern. Testing-effort components such as manpower utilization are dependent on the outcome of testing. Hence how the resources are consumed with each failure and removal attempt is a very important factor during decision making on resource allocation. Therefore we formulate an MTEF that gives a functional relationship between testing and fault removal. The time factor is not explicitly present in the model. Marginal testing effort (MTE) is the amount of effort required to remove an additional fault at any given time. Hence, if m faults have already been removed from the software, the MTE is the testing effort required to remove the $(m+1)$-th fault. We propose a mathematical relationship between the MTE and the number of faults removed based upon the assumption that the MTE is inversely proportional to the remaining faults in the software, i. e. the more faults we remove, the greater effort would be required to remove the next fault.

Mathematically this can be written as

$$w(m) = \frac{k'}{a-m} \tag{25.43}$$

$$\text{and} \quad W(m) = k' \ln \frac{a}{a-m} \,. \tag{25.44}$$

In this expression it is also implicitly assumed that the software contains a finite number of faults at the initiation of testing, that fault removal is perfect and that no new faults are introduced in the process. These assumptions are similar to those used by *Goel* and *Okumoto* [25.10] for their SRGM, i. e. the rate of fault removal is proportional to the remaining faults at any given time. The SRGM is with respect to execution time and it is the mean-value function of the underlying stochastic process described by an NHPP. As the optimization problems being studied in this chapter are with respect to testing resource consumption, MTEF is better suited for our purpose. But the variability in the nature of relationship between the variables i. e. resource consumption and fault removal needs to be captured in a MTEF. It should also include the effect of learning on the testing team. With each additional fault removed, some more faults lying on the execution path are removed and the testing team also gains insight into the software. To incorporate this, we assume that, the MTE is also inversely proportional to a linear function of the number of faults removed. Hence (25.44) can be written as,

$$w(m) = \frac{k}{(p+qm)(a-m)} \tag{25.45}$$

$$\text{and} \quad W(m) = \frac{k}{p+aq} \ln \frac{a(p+qm)}{p(a-m)} \,. \tag{25.46}$$

A higher value of q denotes a higher rate of learning of the testing team and implies a growth rate in the MTE. In expression (25.45) it is assumed that, on a failure, the fault causing that failure is immediately removed with unit probability. Though every care is taken to correct the cause of a failure, the possibility of imperfect debugging and fault generation cannot be ruled out [25.3, 6, 40]. If the fault remains, even after debugging, then it is said to be imperfectly debugged. New faults can also be introduced during the removal process. In both ways the fault content enhances the chance of failure in future. As this phenomenon is a reality, the MTEF should ideally contain the effect of imperfect debugging and fault generation. We modify (25.45) through the assumption that the number of faults imperfectly debugged and generated during the debugging phenomenon is dependent

upon the number of removal attempts already made. The following expression results from these assumptions:

$$w(m) = \frac{k}{(p+qm)} \frac{1}{(a+\delta m - m)} = \frac{k}{(p+qm)} \frac{1}{(a-\gamma m)}, \tag{25.47}$$

$$W(m) = \frac{k}{p\gamma + aq} \ln \frac{a(p+qm)}{p(a-\gamma m)}. \tag{25.48}$$

The modules of a piece of software are independent pieces of software themselves. Hence the least-squares method suggested above can be used to estimate the parameters of the MTEFs of the different modules. For this it is required that modules have already been tested for some time and data pertaining to failures and resource consumption has been recorded.

25.4.3 Optimization

During module testing, modules are tested independently, i. e. the testing teams are different. Again each module can be visualized as independent software and hence distinct MTEFs can be used to describe their testing resource consumption. After the modules have been tested for some time, the parameters of the MTEF viz. a_i, p_i, q_i, k_i can be estimated. Based upon these estimates optimal allocation of resources among modules can be calculated.

Using the MTEF (25.44), the optimization problem (25.41) can be formulated as

$$\text{maximize} \sum_{i=1}^{N} m_i$$

$$\text{subject to} \sum_{i=1}^{N} c_i k_i \ln \frac{a_i}{a_i - m_i} = B. \tag{25.49}$$

We can solve this problem by the method of Lagrangian multipliers. Defining the Lagrange function as

$$L(m_1, m_2, \ldots, m_N, \theta) = \sum_{i=1}^{N} m_i - \theta \left(\sum_{i=1}^{N} c_i \int_0^{m_i} \frac{k_i}{a_i - x} \mathrm{d}x - B \right), \tag{25.50}$$

we get the following optimality conditions

$$\frac{c_i k_i}{a_i - m_i} = \text{constant} \quad \forall i \tag{25.51}$$

$$\text{and} \sum_{i=1}^{N} c_i k_i \ln \frac{a_i}{a_i - m_i} = B. \tag{25.52}$$

After some algebraic simplifications, from (25.52) and (25.52) we obtain

$$W_i^* = \frac{B - \ln \prod_{i=1}^{N} \left(\frac{a_j d_i}{d_j a_i} \right)^{d_j}}{\sum_{j=1}^{N} \frac{d_j}{d_i}}, \quad i = 1, \ldots, N. \tag{25.53}$$

Which is the optimal allocation of testing resources for the *i-th* module in terms of B, a_i and d_i. We have used here the simplest among the MTEFs proposed above. Though obtaining closed-form solution such as (25.53) is always desirable, arriving at one becomes nearly impossible if (25.41) is made more complex. Even with the other two MTEFs in (25.45) and (25.47) the method of Lagrangian multipliers does not directly provide a solution. As the objective here is to highlight the use of marginal effort modeling in allocation problems, we formulate these optimization problems as nonlinear programming problems that could be solved by any of the known methods.

In the solution obtained through (25.53) some modules can receive zero allocations. Hence the minimum number (percentage) of faults that are desired to be removed from each module should be added as a constraint, as in (25.42). Consider the following optimization problem where (25.45) and (25.47) have been substituted into the problem (25.42).

$$\text{Maximize} \sum_{i=1}^{N} \alpha_i m_i$$

subject to

$$m_i \geq r_i, \quad i = 1, \ldots, N,$$

$$\sum_{i=1}^{N} \frac{c_i k_i}{p_i + a_i q_i} \ln \frac{a_i(p_i + q_i m_i)}{p_i(a_i - m_i)} = B. \tag{25.54}$$

With the MTEF (25.47) the resource constraint takes the following form (25.55), the objective function and the other constraint remaining the same in the problem:

$$\sum_{i=1}^{N} \frac{c_i k_i}{p_i \gamma_i + a_i q_i} \ln \frac{a_i(p_i + q_i m_i)}{p_i(a_i - \gamma_i m_i)} = B. \tag{25.55}$$

Equations (25.54) and (25.55) are nonlinear programming problems and any of the standard methods can be used to solve them. But when the number of modules increases, deriving the solution manually becomes difficult. We have solved the problem above with

the help of a software a packages for higher numbers of modules. Once the optimal m_i are found, they can be substituted into (25.44) or (25.48) to find the optimal allocation of resources to the modules. Optimization techniques such as dynamic programming and fuzzy mathematical programming have also been used by the authors for solving more complex resource-allocation problems [25.20, 41].

References

25.1 X. Zhang, H. Pham: An analysis of factors affecting software reliability, J. Syst. Softw. **50**, 43–56 (2000)

25.2 S. Bittanti, P. Bolzern, E. Pedrotti, R. Scattolini: *A flexible modelling approach for software reliability growth*, ed. by G. Goos, J. Harmanis (Springer Verlag, Berlin Heidelberg New York 1988) pp. 101–140

25.3 P. K. Kapur, R. B. Garg, S. Kumar: *Contributions to hardware and software reliability* (World Scientific, Singapore 1999)

25.4 P. K. Kapur, A. K. Bardhan, O. Shatnawi: On why software reliability growth modeling should define errors of different severity, J. Indian Stat. Assoc. **40(2)**, 119–142 (2002)

25.5 M. R. Lyu (Ed.): *Handbook of Software Reliability Engineering* (McGraw Hill, New York 1996)

25.6 J. D. Musa, A. Iannino, K. Okumoto: *Software Reliability: Measurement, Prediction, Applications* (McGraw Hill, New York 1987)

25.7 M. Ohba: Software reliability analysis models, IBM J. Res. Dev., **28**, 428–443 (1984)

25.8 H. Pham: *Software Reliability* (Springer Verlag, Singapore 2000)

25.9 M. Xie: *Software reliability modelling* (World Scientific, Singapore 1991)

25.10 A. L. Goel, K. Okumoto: Time dependent error detection rate model for software reliability and other performance measures, IEEE Trans. Reliab. R **28(3)**, 206–211 (1979)

25.11 W. D. Brooks, R. W. Motley: *Analysis of discrete software reliability models, Technical Report RADC-TR-80-84* (Rome Air Development Center, New York 1980)

25.12 T. Dohi, Y. Nishio, S. Osaki: Optimal software release scheduling based on artificial neural networks, Ann. Softw. Eng. **8**, 167–185 (1999)

25.13 P. K. Kapur, R. B. Garg: Optimal software release policies for software reliability growth models under imperfect debugging, Recherche Operationnelle – Oper. Res. **24(3)**, 295–305 (1990)

25.14 M. Kimura, T. Toyota, S. Yamada: Economic analysis of software release problems with warranty cost and reliability requirement, Reliab. Eng. Syst. Safety **66**, 49–55 (1999)

25.15 S. Yamada, S. Osaki: Optimal software release policies with simultaneous cost and reliability requirements, Eur. J. Oper. Res. **31(1)**, 46–51 (1987)

25.16 B. Yang, M. Xie: A study of operational and testing reliability in software reliability analysis, Reliab. Eng. Syst. Safety **70**, 323–329 (2000)

25.17 A. K. Bardhan: Modelling in software reliability and its interdisciplinary nature. Ph.D. Thesis (Univ. of Delhi, Delhi 2002)

25.18 P. Kubat, H. S. Koch: Managing test procedures to achieve reliable software, IEEE Trans. Reliab. **39(2)**, 171–183 (1993)

25.19 S. Yamada, H. Ohtera: Software reliability growth model for testing effort control, Eur. J. Oper. Res. **46**, 343–349 (1990)

25.20 P. K. Kapur, P. C. Jha, A. K. Bardhan: Optimal allocation of testing resource for a modular software, Asia Pac. J. Oper. Res. **21(3)**, 333–354 (2004)

25.21 G. Q. Kenny: Estimating defects in a commercial software during operational use, IEEE Trans. Reliab. **42(1)**, 107–115 (1993)

25.22 F. M. Bass: A new product growth model for consumer durables, Man. Sci. **15(5)**, 215–224 (1969)

25.23 M. Givon, V. Mahajan, E. Muller: Software piracy: estimation of lost sales and the impact on software diffusion, J. Market. **59**, 29–37 (1995)

25.24 S. Inoue, S. Yamada: Testing-coverage dependent software reliability growth modeling, Int. J. Qual. Reliab. Safety Eng. **11**(4), 303–312 (2004)

25.25 H. Pham, X. Zhang: NHPP software reliability and test models with testing coverage, Eur. J. Oper. Res. **145**, 443–454 (2003)

25.26 H. Yamada, H. Ohtera, H. Narihisa: Software reliability growth models with testing effort, IEEE Trans. Reliab. **R-35(1)**, 19–23 (1986)

25.27 M. Trachtenberg: A general theory of software reliability modeling, IEEE Trans. Reliab. **39(1)**, 92–96 (1990)

25.28 C-Y. Huang, S-Y. Kuo, J. Y. Chen: Analysis of a software reliability growth model with logistic testing effort function, Proc. 8th Int. Symp. Softw. Reliab. Eng., November 1997, pp. 378–388

25.29 S. Yamada, J. Hishitani, S. Osaki: Software-reliability growth model with a Weibull test effort: a model and application, IEEE Trans. Reliab. **42(1)**, 100–106 (1993)

25.30 H. Pham, L. Nordmann, X. Zhang: A general imperfect software-debugging model with S-shaped fault detection rate, IEEE Trans. Reliab. **R-48**, 169–175 (1999)

25.31 P. K. Kapur, R. B. Garg: A software reliability growth model for an error removal phenomenon, Softw. Eng. J. **7**, 291–294 (1992)

25.32 www.dacs.dtic.mil: Software reliability data; Data and Analysis Center for software, USA

25.33 H. Pham: A software cost model with warranty and risk costs, IEEE Trans. Comput. **48(1)**, 71–75 (1999)

25.34 P. K. Kapur, R. B. Garg, V. K. Bahlla: Release policies with random software life cycle and penalty cost, Microelectr. Reliab. **33(1)**, 7–12 (1993)

25.35 P. K. Kapur, R. B. Garg: Cost–reliability optimum release policies for software system under penalty cost, Int. J. Syst. Sci **20**, 2547–2562 (1989)

25.36 S. Yamada: Software reliability measurement during operational phase and its application, J. Comput. Softw. Eng. **1(4)**, 389–402 (1993)

25.37 S. Yamada, M. Ohba, S. Osaki: S-shaped software reliability growth modelling for software error detection, IEEE Trans. Reliab. **R-32(5)**, 475–484 (1983)

25.38 P. K. Kapur, A. K. Bardhan: Testing effort control through software reliability growth modelling, Int. J. Modelling Simul. **22(1)**, 90–96 (2002)

25.39 P. Kubat: Assessing reliability of modular software, Oper. Res. Lett. **8**, 35–41 (1989)

25.40 S. Yamada, K. Tokuno, S. Osaki: Imperfect debugging models with fault introduction rate for software reliability assessment, Int. J. Syst. Sci. **23(2)**, 2241–2252 (1992)

25.41 P. K. Kapur, P. C. Jha, A. K. Bardhan: Dynamic programming approach to testing resource allocation problem for modular software, J. Ratio Math. **14**, 27–40 (2003)

26. An Experimental Study of Human Factors in Software Reliability Based on a Quality Engineering Approach

In this chapter, we focus on a software design-review process which is more effective than other processes for the elimination and prevention of software faults in software development. Then, we adopt a quality engineering approach to analyze the relationships among the quality of the design-review activities, i.e., software reliability, and human factors to clarify the fault-introduction process in the design-review process.

We conduct a design-review experiment with graduate and undergraduate students as subjects. First, we discuss human factors categorized as predispositions and inducers in the design-review process, and set up controllable human factors in the design-review experiment. In particular, we lay out the human factors on an orthogonal array based on the method of design of experiments. Second, in order to select human factors that affect the quality of the design review, we perform a software design-review experiment reflecting an actual design process based on the method of design of experiments. To analyze the experimental results, we adopt a quality engineering approach, i.e., the Taguchi method. That is, applying the orthogonal array $L_{18}(2^1 \times 3^7)$ to the human-factor experiment, we carry out an analysis of variance by using the signal-to-noise ratio (SNR), which can evaluate the stability of the quality characteristics, discuss effective human factors, and obtain the optimal levels for the selected predispositions and inducers.

Further, classifying the faults detected by design-review work into descriptive-design and symbolic-design faults, we discuss the relationships among them in more detail.

Software faults introduced by human errors in development activities of complicated and diverse software systems have resulted in many system failures in modern computer systems. Since these faults are related to the mutual relations among human factors in such software development projects, it is difficult to prevent such software failures beforehand in software production control. Additionally, most of these faults are detected and corrected after software failure occurrences during the testing phase.

If we can make the mutual relations among human factors [26.1, 2] clear, then we expect the problem of software reliability improvement to be solved. So far, several studies have been carried out to investigate the relationships among software reliability and human factors by performing software development experiments and providing fundamental frameworks for understand-

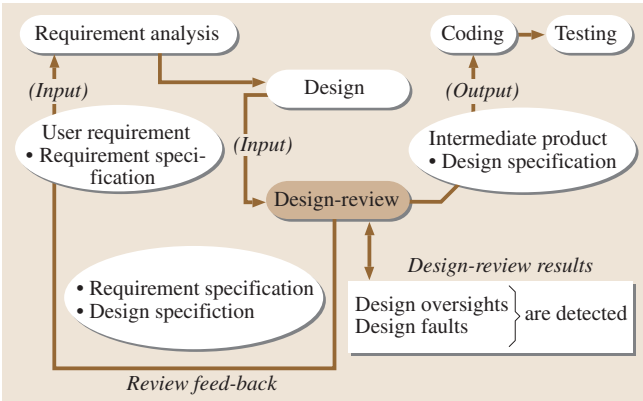


Fig. 26.1 Inputs and outputs in the software design process

ing the mutual relations among various human factors; see [26.3, 4].

In this chapter, we focus on a software design-review process that is more effective than other processes for the elimination and prevention of software faults. Then, we adopt a quality engineering approach [26.5, 6] to analyze the relationships among the quality of the design-review activities, i. e., software reliability, and human factors to clarify the fault-introduction process in the design-review process.

Furthermore, classifying the faults detected by the design-review work into descriptive-design and symbolical-design faults, we discuss the relationships among them.

26.1 Design Review and Human Factors

26.1.1 Design Review

The inputs and outputs for the design-review process are shown in Fig. 26.1 The design-review process is the intermediate process between the design and coding phases, and has software requirement specifications as inputs and software design specifications as outputs. In this process, software reliability is improved by detecting software faults effectively [26.7].

26.1.2 Human Factors

The attributes of the software designers and the design-process environment are related through the design-review process. Therefore, human factors that influence the design specification are classified into two kinds of attributes as follows [26.8–11] (Fig. 26.2):

1. Attributes of the design reviewers (predispositions)
   The attributes of the design reviewers are those of the software engineers who are responsible for the design-review work, for example, the degree of understanding of the requirement specifications and design methods, the aptitude of the programmers, their experience of and capability for software design, the volition of achievement of software design, etc. Most of these are psychological human factors which are considered to contribute directly to the quality of software design specification.
2. Attributes of the design-review environment (inducers)
   In terms of design-review work, many kinds of influential factors are considered, such as the learning level of design methods, the type of design methodologies, physical environmental factors for the software design work, e.g., temperature, humidity, noise, etc. All of these influential factors may indirectly affect the quality of the software design specification.

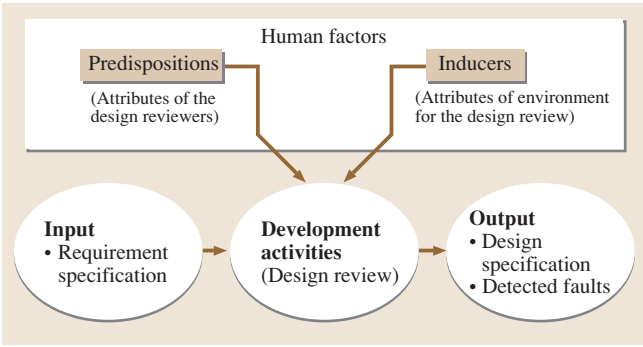


Fig. 26.2 A human-factor model including the predispositions and inducers

26.2 Design-Review Experiment

26.2.1 Human Factors in the Experiment

In order to discover the relationships between the reliability of the software design specification and the human factors that influence it, we have performed a design-review experiment by selecting five human factors, as shown in Table 26.1, as control factors concerned with the review work.

- BGM (background music) of classical music in the review-work environment (inducer A)
  Design-review work for detecting faults requires concentrated attentiveness. We adopt a BGM of classical music as a factor of the work environment that maintains review efficiency.
- Time duration of design-review work (inducer B)
  In this experiment, we set the subjects design-review work to be completed in approximately 20 min. We adopt three time durations for software design-review work, such as 20 min, 30 min and 40 min.
- Check list (inducer E)
  We prepare a check list (CL), which indicates the matters to be noticed in the review work. This factor has the following three levels: detailed CL, common CL, and without CL.
- Degree of understanding of the design method (predisposition C)
  Predisposition C of the two predispositions is the degree of understanding of the design method R-Net (requirements network). Based on preliminary tests of the ability to understand the R-Net technique, the subjects are divided into the following three groups: high, common, and low ability.
- Degree of understanding of the requirement specification (predisposition D)
  Predisposition D of the two predispositions is the degree of understanding of the requirement specification. Similarly to predisposition C, based on preliminary tests of geometry ability, the subjects are divided into the following three groups: high, common, and low ability.

26.2.2 Summary of Experiment

In this experiment, we conduct an experiment to clarify the relationships among human factors affecting software reliability and the reliability of design-review work by assuming a human-factor model consisting of predispositions and inducers, as shown in Fig. 26.2. The actual experiment has been performed by 18 subjects based on the same design specification of a triangle program which receives three integers representing the sides of a triangle and classifies the kind of triangle that these sides form [26.12]. We measured the capability of the 18 subjects in terms of their degree of understanding of the design method and the requirement specification by using preliminary tests before the design of experiment. Furthermore, we seeded some faults in the design specification intentionally. We then executed this design-review experiment in which the 18 subjects detected the seeded faults.

We performed the experiment using the five control factors with three levels, as shown in Table 26.1, which are assigned to the orthogonal array $L_{18}(2^1 \times 3^7)$ of the design of experiment, as shown in Table 26.3.

Table 26.1 Controllable factors in the design-review experiment

	Control factor	Level		
		1	2	3
A	BGM of classical music to review-work environment (inducer)	A_1: yes	A_2: no	–
B	Time duration of design-review work (minute) (inducer)	B_1: 20 min	B_2: 30 min	B_3: 40 min
C	Degree of understanding of the design method (R-Net technique) (predisposition)	C_1: high	C_2: common	C_3: low
D	Degree of understanding of the requirement specification (predisposition)	D_1: high	D_2: common	D_3: low
E	Check list (indicating the matters that require attention in the review work) (inducer)	E_1: detailed	E_2: common	E_3: nothing

26.3 Analysis of Experimental Results

26.3.1 Definition of SNR

We define the efficiency of the design review, i. e., the reliability, as the degree that the design reviewers can accurately detect correct and incorrect design parts for a design specification containing seeded faults. There exists the following relationship among the total number of design parts, n, the number of correct design parts, n_0, and the number of incorrect design parts containing seeded faults, n_1:

$$n = n_0 + n_1 \,. \tag{26.1}$$

Therefore, the design parts are classified as shown in Table 26.2 by using the following notation:

n_{00} = the number of correct design parts detected accurately as correct design parts,
n_{01} = the number of correct design parts detected by mistake as incorrect design parts,
n_{10} = the number of incorrect design parts detected by mistake as correct design parts,
n_{11} = the number of incorrect design parts detected accurately as incorrect design parts,

Table 26.2 Input and output tables for the two kinds of error

(i) Observed values

Input \ Output	0 (true)	1 (false)	Total
0 (true)	n_{00}	n_{01}	n_0
1 (false)	n_{10}	n_{11}	n_1
Total	r_0	r_1	n

(ii) Error rates

Input \ Output	0 (true)	1 (false)	Total
0 (true)	$1-p$	p	1
1 (false)	q	$1-q$	1
Total	$1-p+q$	$1-q+p$	2

where the two kinds of error rate are defined by

$$p = \frac{n_{01}}{n_0} \,, \tag{26.2}$$

$$q = \frac{n_{10}}{n_1} \,. \tag{26.3}$$

Table 26.3 Controllable factors in the design-review experiment

Experiment	Control factors								Observed values				SNR (dB)
						Error							
No.	*A*	*B*	*C*	*D*	*E*	*e*	*e*	*e*	n_{00}	n_{01}	n_{10}	n_{11}	
1	1	1	1	1	1	1	1	1	110	1	2	16	8.404
2	1	1	2	2	2	2	2	2	108	3	10	8	−0.515
3	1	1	3	3	3	3	3	3	109	2	16	2	−6.050
4	1	2	1	1	2	2	3	3	111	0	2	16	10.008
5	1	2	2	2	3	3	1	1	107	4	4	14	2.889
6	1	2	3	3	1	1	2	2	104	7	11	7	−4.559
7	1	3	1	2	1	3	2	3	111	0	4	14	8.104
8	1	3	2	3	2	1	3	1	106	5	8	10	−0.780
9	1	3	3	1	3	2	1	2	110	1	11	7	2.099
10	2	1	1	3	3	2	2	1	110	1	11	7	2.099
11	2	1	2	1	1	3	3	2	106	5	4	14	2.260
12	2	1	3	2	2	1	1	3	105	6	12	6	−4.894
13	2	2	1	2	3	1	3	2	105	6	10	8	−2.991
14	2	2	2	3	1	2	1	3	108	3	15	3	−5.784
15	2	2	3	1	2	3	2	1	105	6	10	8	−2.991
16	2	3	1	3	2	3	1	2	109	2	2	16	6.751
17	2	3	2	1	3	1	2	3	107	4	4	14	2.889
18	2	3	3	2	1	2	3	1	103	8	9	9	−3.309

Considering the two kinds of error rate, p and q, we can derive the standard error rate, p_0 [26.6] as

$$p_0 = \frac{1}{1+\sqrt{\left(\frac{1}{p}-1\right)\left(\frac{1}{q}-1\right)}} \,. \tag{26.4}$$

Then, the signal-to-noise ratio based on (26.4) is defined by [26.6]

$$\eta_0 = -10\log_{10}\left(\frac{1}{(1-2p_0)^2}-1\right) \,. \tag{26.5}$$

The standard error rate, p_0, can be obtained from transforming (26.5) by using the signal-to-noise ratio of each control factor as

$$p_0 = \frac{1}{2}\left(1-\frac{1}{\sqrt{10^{(-\frac{\eta_0}{10})}+1}}\right) \,. \tag{26.6}$$

26.3.2 Orthogonal Array $L_{18}(2^1 \times 3^7)$

The method of experimental design based on orthogonal arrays is a special one that requires only a small number of experimental trials to help discover the main factor effects. In traditional research [26.4, 8], the design of experiment has been conducted by using the orthogonal array $L_{12}(2^{11})$. However, since the orthogonal array $L_{12}(2^{11})$ is applied for grasping the factor effect between two levels the human factors experiment, the middle effect between two levels cannot be measured. Thus, in order to measure it, we adopt the orthogonal array $L_{18}(2^1 \times 3^7)$, which can lay out one factor with two levels (1, 2) and seven factors with three levels (1, 2, 3), as shown in Table 26.3, and dispense with $2^1 \times 3^7$ trials by executing 18 experimentally independent experimental trials each other. For example, as for experimental trial no. 10, we executed the design-review work under the conditions A_2, B_1, C_1, D_3, and E_3, and obtained a computed SNR of 2.099 dB from the observed values $n_{00} = 110$, $n_{01} = 1$, $n_{10} = 11$, and $n_{11} = 7$.

Additionally, the interaction between two factors can be estimated without sacrificing a factor. Any pair of human factors are partially mixed with the effect of the remaining factors. Therefore, we have evaluated the large effects of highly reproducible human factors because the selected optimal levels of the relatively large factor has a larger effect than that of the smaller one.

Considering these circumstances, we can obtain the optimal levels for the selected inhibitors and inducers efficiently by using the orthogonal array $L_{18}(2^1 \times 3^7)$.

26.4 Investigation of the Analysis Results

26.4.1 Experimental Results

The experimental results for the observed values of the design parts discussed in Sect. 26.3.1 in the software design specification are shown in Table 26.3. The SNR data calculated using (26.5) are also shown in Table 26.3.

26.4.2 Analysis of Variance

The result of the analysis of variance for the observed correct and incorrect design parts is shown in Table 26.4 by using the SNR data, as shown in Table 26.3. In Table 26.4, f, S, V, F_0, and ρ represent the degree of freedom, the sum of squares, the unbiased variance, the unbiased variance ratio, and the contribution ratio, respectively, for performing the analysis of variance. In order to obtain the precise analysis results, the check list factor (factor E) is pooled with the error factor (factor e). We then performed the analysis of variance based on the new pooled error factor (factor e').

26.4.3 Discussion

As a result of the experimental analysis, the effective control factors such as the BGM of classical music to review-work environment (factor A), the duration of the design-review work (factor B), the degree of understanding of the software design method (Factor C), and the degree of understanding of the requirement specification (factor D) were recognized. In particular, factors A and B are mutually interacting.

We then find that our experience from actual software development [26.8] and the experimental result above based on a design review are equivalent. Table 26.5 shows the comparisons of SNRs and standard error rates. The improvement ratio of the reliability of design review is calculated as 20.909 dB [i. e. 33.1% measured in the standard error rate in (26.4) from (26.5)] by using the SNR based on the optimal condition (A_1, B_3, C_1, D_1) of the control factors, such as A, B, C, and D, whose effects are rec-

Table 26.4 The result of analysis of variance using the SNR

Factor	f	S	V	F_0	ρ (%)
A	1	36.324	36.324	10.578*	7.4
B	2	33.286	16.643	4.847*	5.9
C	2	229.230	114.615	33.377**	49.8
D	2	86.957	43.479	12.661**	17.9
E	2	3.760	1.880○		
$A \times B$	2	33.570	16.785	4.888*	6.0
e	6	23.710	3.952○	–	
e'	8	27.470	3.434	–	13.0
T	17	446.837	–	–	100.0

○: pooled, *: 5% level of significance,
**: 1% level of significance

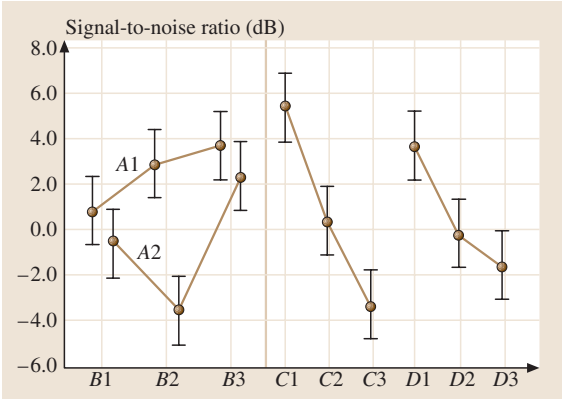


Fig. 26.3 Estimation of significant factors

ognized in Fig. 26.3. Therefore, it is expected that a quantitative improvement in the reliability of design review can be achieved by using these control factors.

26.5 Confirmation of Experimental Results

Table 26.6 shows the optimal and worst levels of the control factors for the design-review discussed in Sect. 26.4. Considering the circumstances, we conduct an additional experiment to confirm the experimental results using the SNR.

26.5.1 Additional Experiment

We focus on the effect of faults detected under the optimal conditions of the design-review work. As for the design of experiment discussed in Sect. 26.2, the design specification is for the triangle program reviewed by 18 subjects. We measured both their degree of understanding of the design method and their degree of understanding of the requirement specification by preliminary tests before the design of the additional experiment.

We also seeded some faults into the design specification intentionally. We then executed the same design-review experiment discussed in Sect. 26.2.2 under the same review conditions (the optimal levels for the selected predispositions). Additionally, we confirmed that the selected predispositions divided by the preliminary tests were consistent with the optimal levels of the two inducers.

The experimental results for the observed values of correct and incorrect design parts and the preliminary tests are shown in Table 26.7 with the SNR data calculated using (26.5).

26.5.2 Comparison of Factorial Effects Under Optimal Inducer Conditions

Figure 26.4 shows the optimal levels of the control factors of the design review based on the additional experiment. If both inhibitors are at the high state, the fault-detection rate is improved. Additionally, Table 26.8 shows a comparison of the SNRs and standard error rates between the optimal levels for the selected inducers. The improvement ratio of the reliability of the design review

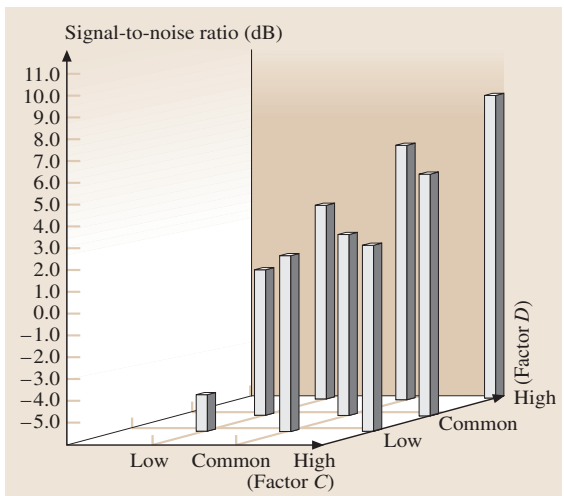


Fig. 26.4 The comparison of factorial effects

Table 26.5 The comparison of SNR and standard error rates

	Optimal conditions (A_1, B_3, C_1, D_1)	Worst conditions (A_2, B_2, C_3, D_3)	Deviation
Signal-to-noise ratio (dB)	10.801	−10.108	Δ 20.909
Confidence interval	±3.186		
Standard error rates (%)	2.0	35.1	Δ 33.1

Table 26.6 The optimal and worst levels of design review

	Control factor	Level	
		Optimal	Worst
Inducer A	BGM of classical music to review-work environment	A_1: yes	A_2: no
Inducer B	Time duration of design-review work (minute)	B_3: 40 min	B_2: 30 min
Predisposition C	Degree of understanding of the design method (R-Net technique)	C_1: high	C_3: low
Predisposition D	Degree of understanding of the requirement specification	D_1: high	D_3: low

Table 26.7 The SNRs in the optimal levels for the selected inducers

No.	Observed values				SNR	Standard	Observed values	
	n_{00}	n_{01}	n_{10}	n_{11}	(dB)	error rates	Factor C	Factor D
1	109	2	3	15	5.613	0.027	High	Common
2	111	0	5	13	7.460	0.040	Common	High
3	108	3	2	16	3.943	0.078	High	Low
4	107	4	4	14	2.889	0.094	High	Low
5	111	0	2	16	10.008	0.023	High	High
6	109	2	3	15	5.613	0.057	Low	High
7	107	4	4	14	2.889	0.094	Common	Low
8	107	4	4	14	2.889	0.094	Low	Common
9	111	0	2	16	10.008	0.023	High	High
10	109	2	4	14	4.729	0.068	Low	High
11	107	4	3	15	3.825	0.080	Common	Common
12	107	4	6	12	1.344	0.120	Low	Common
13	101	10	8	10	−3.385	0.220	Low	Low
14	105	6	3	15	2.707	0.097	Common	Low
15	107	4	3	16	3.825	0.080	Common	Common
16	111	0	4	14	8.104	0.035	Common	High
17	111	0	5	13	7.460	0.040	High	Common
18	98	13	9	9	−3.369	0.025	Low	Low

Table 26.8 The comparison of SNRs and standard error rates between the optimal levels for the selected inducers

	Factor C and factor D		Deviation
	High	Low	
Signal-to-noise ratio (dB)	10.008	−3.510	Δ 13.518
Standard error rates (%)	2.3	22.3	Δ 20.0

is calculated as 13.518 dB (i. e. 20.0% measured in the standard error rate) by using the signal-to-noise ratio based on the optimal conditions of the control factors, such as A, B, C, and D, whose effects are recognized in Fig. 26.4. Thus, we can confirm that the optimal levels of the two inducers are consistent with the optimal levels of the two predispositions C and D divided by the preliminary tests.

26.6 Data Analysis with Classification of Detected Faults

26.6.1 Classification of Detected Faults

We can distinguish the design parts as follows to be pointed out in the design review as detected faults into descriptive-design and symbolical-design parts, denoted by R_1 and R_2, respectively.

- Descriptive-design faults
  The descriptive-design parts consist of words or technical terminologies which are described in the design specification to realize the required functions. In this experiment, the descriptive design faults are algorithmic, and we can improve the quality of the design specification by detecting and correcting them.
- Symbolical-design faults
  The symbolical-design parts consist of marks or symbols which are described in the design specification. In this experiment, the symbolical-design faults are notation mistakes, and the quality of the design specification cannot be improved by detecting and correcting them.

26.6.2 Data Analysis

The experimental results for the observed values classified into the two types of design parts discussed in Sect. 26.6.1 are shown in Table 26.9. The SNR data calculated through (26.5) are also shown in Table 26.9.

A result of the analysis of variance for the descriptive-design parts is shown in Table 26.10, and that for the symbolical-design parts shown in Table 26.11, based on the SNR data shown in Table 26.9. Figure 26.5 shows the effect for each level in the control factor that affects the design-review result based on the SNR calculated from the observation values.

Descriptive-Design Faults

In design-review work, the effective review conditions for correcting and removing algorithmic faults are BGM of classical music, "yes(A_1)" and design-review time, "40 minutes(B_3)". The reviewer's capability, the degree of understanding of the design method (R-Net Technique), "high(C_1)", and that of the require-

Table 26.9 The orthogonal array $L_{18}(2^1 \times 3^7)$ with assigned human factors and experimental data

No.	Control factor					Observed values								SNR (db)	
						Descriptive-design parts R_1				Symbolic-design parts R_2					
	A	B	C	D	E	n_{00}	n_{01}	n_{10}	n_{11}	n_{00}	n_{01}	n_{10}	n_{11}	R_1	R_2
1	1	1	1	1	1	52	0	2	12	58	1	0	4	7.578	6.580
2	1	1	2	2	2	49	3	8	6	59	0	2	2	−3.502	3.478
3	1	1	3	3	3	50	2	12	2	59	0	4	0	−8.769	−2.342
4	1	2	1	1	2	52	0	2	12	59	0	0	4	7.578	8.237
5	1	2	2	2	3	50	2	4	10	57	2	0	4	1.784	4.841
6	1	2	3	3	1	45	7	8	6	59	0	3	1	−7.883	0.419
7	1	3	1	2	1	52	0	2	12	59	0	2	2	7.578	3.478
8	1	3	2	3	2	47	5	6	8	59	0	2	2	−3.413	3.478
9	1	3	3	1	3	52	0	10	4	58	1	1	3	0.583	4.497
10	2	1	1	3	3	52	0	10	4	58	1	1	3	0.583	4.497
11	2	1	2	1	1	47	5	1	13	59	0	3	1	3.591	0.419
12	2	1	3	2	2	46	6	8	6	59	0	4	0	−6.909	−2.342
13	2	2	1	2	3	46	6	10	4	59	0	0	4	−10.939	8.237
14	2	2	2	3	1	49	3	11	3	59	0	4	0	−8.354	−2.342
15	2	2	3	1	2	46	6	10	4	59	0	0	4	−10.939	8.237
16	2	3	1	3	2	50	2	2	12	59	0	0	4	4.120	8.237
17	2	3	2	1	3	50	2	4	10	57	2	0	4	1.784	4.841
18	2	3	3	2	1	44	8	6	8	59	0	3	1	−5.697	0.419

Table 26.10 The result of analysis of variance (descriptive-design faults)

Factor	f	S	V	F_0	ρ (%)
A	1	65.338	65.338	7.915*	8.071
B	2	96.907	48.454	5.869*	11.367
C	2	263.701	131.851	15.972**	34.950
D	2	108.953	54.477	6.599*	13.070
E	2	13.342	6.671○		
$A \times B$	2	106.336	53.168	6.053*	12.700
e	6	52.699	8.783○	–	
e'	8	66.041	8.255	–	19.842
T	17	707.276	–	–	100.0

○: pooled, *: 5% level of significance,
**: 1% level of significance

Table 26.11 The result of analysis of variance (symbolic-design faults)

Factor	f	S	V	F_0	ρ (%)
A	1	0.037	0.037○		
B	2	29.041	14.521	2.975	8.180
C	2	86.640	43.320	8.875**	32.618
D	2	38.300	43.320	3.923	12.108
E	2	37.783	18.892	3.870	11.889
$A \times B$	2	4.833	2.416○		
e	6	38.759	6.460○	–	
e'	9	43.929	4.881	–	35.206
T	17	235.693	–	–	100.0

○: pooled, *: 5% level of significance,
**: 1% level of significance

ment specification, "high(D_1)", are derived as optimal conditions.

Symbolic-Design Faults

In design-review work, the optimal condition for effective review conditions for correcting and removing notation mistakes is that the degree of understanding of the requirement specification is "high(C_1)".

26.6.3 Data Analysis with Correlation Among Inside and Outside Factors

Furthermore, classifying the detected faults as due to the outside factor R and the inside control factors A, B, C, D, and E, as shown in Table 26.9, we can perform the analysis of variance. Here, the outside factor R has two

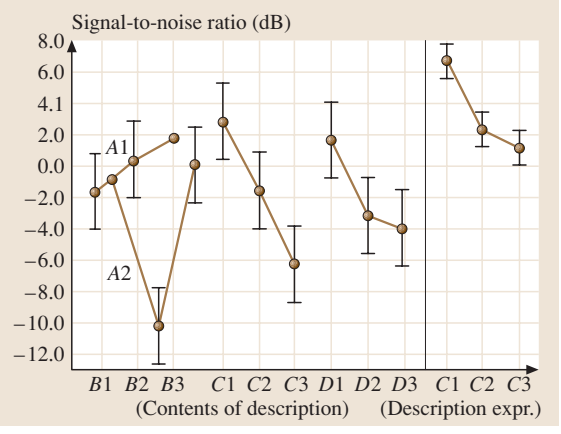


Fig. 26.5 The estimation of significant factors with classification of detected faults

Table 26.12 The result of analysis of variance by taking account of correlation among inside and outside factors

Factor	f	S	V	F_0	ρ (%)
A	1	37.530	37.530	2.497	3.157
B	2	47.500	23.750	1.580	3.995
C	2	313.631	156.816	10.435**	26.380
D	2	137.727	68.864	4.582*	11.584
E	2	4.684	2.342	0.156	0.394
$A \times B$	2	44.311	22.155	1.474	3.727
e_1	6	38.094	6.460	0.422	3.204
R	1	245.941	16.366	16.366**	20.686
$A \times R$	1	28.145	28.145	1.873	2.367
$B \times R$	2	78.447	39.224	2.610	6.598
$C \times R$	2	36.710	18.355	1.221	3.088
$D \times R$	2	9.525	4.763	0.317	0.801
$E \times R$	2	46.441	23.221	1.545	3.906
e_2	8	120.222	15.028	3.870	10.112
T	35	1188.909			100.0

*: 5% level of significance, **: 1% level of significance

levels, corresponding to descriptive-design parts (R_1) and symbolical-design parts (R_2).

As a result of the analysis of variance, by taking account of correlation among inside and outside factors, we can obtain Table 26.12. There are two kinds of errors in the analysis of variance: e_1 is the error among the experiments of the inside factors, and e_2 is the mutual correlation error between e_1 and the outside factor. In this analysis, since there was no significant effect by performing F-testing for e_1 with e_2, F-testing for all factors was performed using e_2. As a result, the significant control factors, such as the degree of understanding of the

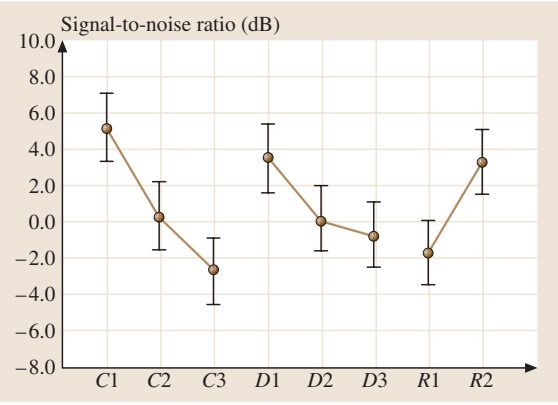


Fig. 26.6 The estimation of significant factors with correlation among inside and outside factors

design method (factor C), the degree of understanding of the requirement specification (factor D), and the classification of the detected faults (factor R), were recognized. Figure 26.6 shows the effect of the factor for each level in the significant factors that affect the design-review work.

As a result of the analysis, in the inside factors, only factors C and D are significant and the inside and outside factors are not mutually interacting. That is, it turns out that the reviewers with a high degree of understanding of the design method and a high degree of understanding of the requirement specification can review the design specification efficiently regardless of the classification of the detected faults. Moreover, the result that the outside factor R is highly significant, i. e., the descriptive-design faults are detected less effectively than the symbolic-design faults, can be obtained. That is, although it is a natural result, it is difficult to detect and correct the algorithmic faults which would lead to an improvement in quality rather than the notation mistakes. However, it is important to detect and correct the algorithmic faults as this is an essential problem for quality improvement in design-review work. Therefore, in order to increase the rate of detection and correction of algorithmic faults, which would lead to quality improvement, before design-review work it is necessary to make reviewers understand fully the design technique used to describe the design specification and the contents of the requirement specifications.

References

26.1 V. R. Basili, R. W. Reiter, Jr.: An investigation of human factors in software development, IEEE Comput. Mag. **12**, 21–38 (1979)

26.2 T. Nakajo, H. Kume: A case history analysis of software error cause–effect relationships, IEEE Trans. Softw. Eng. **17**, 830–838 (1991)

26.3 K. Esaki, M. Takahashi: Adaptation of quality engineering to analyzing human factors in software design, J. Qual. Eng. Forum **4**, 47–54 (1996) (in Japanese)

26.4 K. Esaki, M. Takahashi: A software design review on the relationship between human factors, software errors classified by seriousness, J. Qual. Eng. Forum **5**, 30–37 (1997) (in Japanese)

26.5 G. Taguchi: *A Method of Design of Experiment*, Vol. 1, 2nd edn. (Maruzen, Tokyo 1976) (in Japanese)

26.6 G. Taguchi (Ed.): *Signal-to-Noise Ratio for Quality Evaluation* (Japanese Standards Association, Tokyo 1998) (in Japanese)

26.7 S. Yamada: *Software Reliability Models: Fundamentals and Applications* (JUSE, Tokyo 1994) (in Japanese)

26.8 K. Esaki, S. Yamada, M. Takahashi: A quality engineering analysis of human factors affecting software reliability in software design review process, Trans. IEICE Jpn. **J84-A**, 218–228 (2001) (in Japanese)

26.9 R. Matsuda, S. Yamada: A human factor analysis for software reliability improvement based on aquality engineering approach in design-review process. In: *Proc. 9th ISSAT Int. Conf. Reliab. Qual. Design, Honolulu, Hawaii, USA*, Tech. Dig., ed. by H. Pham, S. Yamada (International Society of Science and Applied Technologies, New Brunswick 2003) pp. 75–79

26.10 S. Yamada, R. Matsuda: A quality engineering evaluation for human factors affecting software reliability in design review process, J. Jpn. Ind. Man. Assoc. **54**, 71–79 (2003) (in Japanese)

26.11 S. Yamada: Recent advances in software reliability modeling. In: *Proc. Intern. Work. Reliab. Appl., Seoul, Korea*, Tech. Dig., ed. by D. H. Park (Korean Reliability Society, Seoul 2003) pp. 19–32

26.12 I. Miyamoto: *Software Engineering – Current Status and Perspectives* (TBS, Tokyo 1982) (in Japanese)

27. Statistical Models for Predicting Reliability of Software Systems in Random Environments

After a brief overview of existing models in software reliability in Sect. 27.1, Sect. 27.2 discusses a generalized nonhomogeneous Poisson process model that can be used to derive most existing models in the software reliability literature. Section 27.3 describes a generalized random field environment (RFE) model incorporating both the testing phase and operating phase in the software development cycle for estimating the reliability of software systems in the field. In contrast to some existing models that assume the same software failure rate for the software testing and field operation environments, this generalized model considers the random environmental effects on software reliability. Based on the generalized RFE model, Sect. 27.4 describes two specific RFE reliability models, the γ-RFE and β-RFE models, for predicting software reliability in field environments. Section 27.4 illustrates the models using telecommunication software failure data. Some further research considerations based on the generalized software reliability model are also discussed.

Many software reliability models have been proposed to help software developers and managers understand and analyze the software development process, estimate the development cost and assess the level of software reliability. Among these software reliability models, models based on the nonhomogeneous Poisson process (NHPP) have been successfully applied to model the software failure processes that possess certain trends such as reliability growth or deterioration. NHPP models are very useful to predict software failures and software reliability in terms of time, and to determine when to stop testing and release the software [27.1].

Currently most existing NHPP software reliability models have been carried out through the fault intensity rate function and the mean-value functions $m(t)$ within a controlled operating environment [27.2–13]. Obviously, different models use different assumptions and therefore provide different mathematical forms for the mean-value function $m(t)$. Table 27.1 shows a summary of several existing models appearing in the software reliability engineering literature [27.14]. Generally, these models are applied to software testing data and then to make predictions of software failures and reliability in the field. The underlying assumption for this application is that the field environments are the same as, or close to, a testing environment; this is valid for some software systems that are only used in one environment throughout their entire lifetime. However, this assumption is not valid for many applications where a software program may be used in many different locations once it is released.

The operating environments for the software in the field are quite different. The randomness of the field environment will affect software failure and software reliability in an unpredictable way. *Yang* and *Xie* [27.15] mentioned that the operational reliability and testing reliability are often different from each other, but they assumed that the operational failure rate is still close to the testing failure rate, and hence that the difference between them is that the operational failure rate decreases with time, while the testing failure rate remains constant. *Zhang* et al. [27.16] proposed an NHPP software reliability calibration model

Table 27.1 Summary of NHPP software reliability models [27.14]

Model name	Model type	MVF [$m(t)$]	Comments
Goel–Okumoto (G–O)	Concave	$m(t) = a(1 - e^{-bt})$ $a(t) = a$ $b(t) = b$	Also called exponential model
Delayed S-shaped	S-shaped	$m(t) = a[1 - (1 + bt)e^{-bt}]$ $a(t) = a$ $b(t) = \frac{b^2 t}{1+bt}$	Modification of G–O model to obtain S-shape
Inflection S-shaped SRGM	Concave	$m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}$ $a(t) = a$ $b(t) = \frac{b}{1+\beta e^{-bt}}$	Solves a technical condition with the G–O model. Becomes the same as G–O if $\beta = 0$
HD/G–O model	Concave	$m(t) = \log\left[(e^a - c)/(e^{a e^{-bt}} - c)\right]$	Same as G–O when $c = 0$
Yamada exponential	Concave	$m(t) = a\left(1 - e^{-r\alpha(1-e^{(-\beta t)})}\right)$ $a(t) = a$ $b(t) = r\alpha\beta e^{-\beta t}$	Attempts to account for testing effort
Yamada Rayleigh	S-shaped	$m(t) = a\left(1 - e^{-r\alpha(1-e^{(-\beta t^2/2)})}\right)$ $a(t) = a$ $b(t) = r\alpha\beta t e^{-\beta t^2/2}$	Attempts to account for testing effort
Yamada imperfect debugging model (1)	S-shaped	$m(t) = \frac{ab}{\alpha+b}(e^{\alpha t} - e^{-bt})$ $a(t) = a e^{\alpha t}$ $b(t) = b$	Assumes exponential fault-content function and constant fault-detection rate
Yamada imperfect debugging model (2)	S-shaped	$m(t) = a(1 - e^{-bt})(1 - \frac{\alpha}{b}) + \alpha a t$ $a(t) = a(1 + \alpha t)$ $b(t) = b$	Assumes constant fault-introduction rate α and constant fault-detection rate
PNZ model	S-shaped and concave	$m(t) = \frac{a}{1+\beta} e^{-bt}\left[(1 - e^{-bt})(1 - \frac{\alpha}{b}) + \alpha a t\right]$ $a(t) = a(1 + \alpha t)$ $b(t) = \frac{b}{1+\beta e^{-bt}}$	Assumes introduction rate is a linear function of testing time, and the fault-detection rate function is nondecreasing and inflexion S-shaped
Pham–Zhang model	S-shaped and concave	$m(t) = \frac{1}{1+\beta e^{-bt}}\left[(c + a)(1 - e^{-bt}) - \frac{a}{b-\alpha}(e^{-\alpha t} - e^{-bt})\right]$ $a(t) = c + a(1 - e^{-\alpha t})$ $b(t) = \frac{b}{1+\beta e^{-bt}}$	Assume constant introduction rate is exponential function of testing time, and the error-detection function is nondecreasing with an inflexion S-shaped model

by introducing a calibration factor. This calibration factor, K, obtained from software failures in both the testing and field operation phases will be a multiplier to the software failure intensity. This calibrated software reliability model can be used to assess and adjust the predictions of software reliability in the operation phase.

Instead of relating the operating environment to the failure intensity λ, in this chapter we assume that the effect of the operating environment is to multiply the unit failure-detection rate $b(t)$ achieved in the testing environment using the concept of the proportional hazard approach suggested by *Cox* [27.17]. If the operating environment is more liable to software failure, then the unit fault-detection rate increases by some factor η greater than 1. Similarly, if the operating environment is less liable to software failure, then the unit fault-detection rate decreases by some positive factor η less than 1.

This chapter describes a model based on the NHPP model framework for predicting software failures and evaluating the software reliability in random field environments. Based on this model, developers and engineers can further develop specific software reliability models customized to various applications.

Notations	
$R(t)$	Software reliability function
η	Random environmental factor
$G(\eta)$	Cumulative distribution function of η
γ	Shape parameter of gamma distributions
θ	Scale parameter of gamma distributions
α, β	Parameters of beta distributions
$N(t)$	Counting process which counts the number of software failures discovered by time t
$m(t)$	Expected number of software failures detected by time t, $m(t) = E[N(t)]$
$a(t)$	Expected number of initial software faults plus introduced faults by time t
$m_1(t)$	Expected number of software failures in testing by time t
$m_2(t)$	Expected number of software failures in the field by time t
$a_1(t)$	Expected number of initial software faults plus introduced faults discovered in the testing by time t
a	Number of initial software faults at the beginning of testing phase, is a *software parameter* that is directly related to the software itself
T	Time to stop testing and release the software for field operations
a_F	Number of initial software faults in the field (at time T)
$b(t)$	Failure detection rate per fault at time t, is a *process parameter* that is directly related to testing and failure process
p	Probability that a fault will be successfully removed from the software
q	Error introduction rate at time t in the testing phase
MLE	Maximum likelihood estimation
RFE-model	Software reliability model subject to a random field environment
γ-RFE	Software reliability model with a gamma distributed field environment
β-RFE	Software reliability model with a beta distributed field environment
NHPP	Non-homogeneous Poisson process
SRGM	software reliability growth model
HD	Hossain–Ram
PNZ	Pham–Nordman–Zhang
G–O	Goel–Okumoto
NHPP	nonhomgeneous Poisson process
MLE	maximum likelihood estimation
RFE	random field environment

27.1 A Generalized NHPP Software Reliability Model

A generalized NHPP model studied by *Zhang* et al.[27.7] can be formulated as follows:

$$m'(t) = \eta b(t)[a(t) - pm(t)] \,, \tag{27.1}$$

$$a'(t) = q \cdot m'(t) \,, \tag{27.2}$$

where $m(t)$ is the number of software failures expected to be detected by time t. If the marginal conditions are given as $m(0) = 0$ and $a(0) = a$, then for a specific environmental factor η, the solutions to (27.1) and (27.2) are, given in [27.7], as follows

$$m_\eta(t) = a \int_0^t \eta b(u) \mathrm{e}^{-\int_0^u \eta(p-q)b(\tau)\mathrm{d}\tau} \mathrm{d}u \,, \tag{27.3}$$

$$a_\eta(t) = a \left[1 + \int_0^t \eta q b(u) \mathrm{e}^{-\int_0^u \eta(p-q)\cdot b(\tau)\mathrm{d}\tau} \mathrm{d}u \right] . \tag{27.4}$$

This is the generalized form of the NHPP software reliability model. When $p = 1$, $\eta = 1$ and $q = 0$, then for any given function $a(t)$ and $b(t)$, all the functions listed in Table 27.1 can easily be obtained.

27.2 Generalized Random Field Environment (RFE) Model

The testing environment is often a controlled environment with much less variation compared to the field environments, which may be quite different for the field application software. Once a software program is released, it may be used in many different locations and various applications in industries. The operating environments for the software are quite different. Therefore, the randomness of the field environment will affect the cumulative software failure data in an unpredictable way.

Figure 27.1 shows the last two phases of the software life cycle: in-house testing and field operation [27.18]. If T is the time to stop testing and release the software for field operations, then the time period $0 \le t \le T$ refers to the time period of *software testing*, while the time period $T \le t$ refers to the post-release period—*field operation*.

The environmental factor η is used to capture the uncertainty about the environment and its effects on the software failure rate. In general, software testing is carried out in a controlled environment with very small variations, which can be used as a reference environment where η is constant and equal to 1. For the field operating environment, the environmental factor η is assumed to be a nonnegative random variable (RV) with probability density function (PDF) $f(\eta)$, i. e.

$$\eta = \begin{cases} 1 & t \le T \\ \text{RV with PDF } f(\eta) & t \ge T \end{cases} . \qquad (27.5)$$

If the value of η is less than 1, this indicates that the conditions are less favorable to fault detection than that of testing environment. Likewise, if the value of η is greater than 1, it indicates that the conditions are more favorable to fault detection than that of the testing environment.

From (27.3) and (27.5), the mean-value function and the function $a_1(t)$ during testing can be obtained as

$$m_1(t) = a \int_0^t b(u) \mathrm{e}^{-\int_0^u (p-q)b(\tau)\mathrm{d}\tau} \mathrm{d}u \quad t \le T ,$$

$$a_1(t) = a\left[1 + \int_0^t qb(u) \times \mathrm{e}^{-\int_0^u (p-q)\cdot b(\tau)\mathrm{d}\tau} \mathrm{d}u\right] \quad t \le T . \qquad (27.6)$$

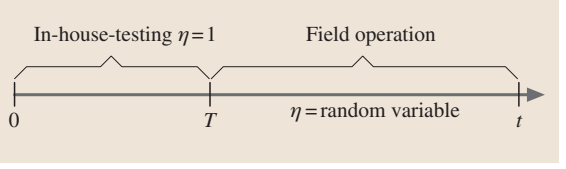


Fig. 27.1 Testing versus field environment where T is the time to stop testing and release the software

For the field operation where $t \ge T$ the mean-value function can be represented as

$$\begin{aligned} m_2(t) &= m_1(T) + \int_0^\infty m_\eta(t) f(\eta)\mathrm{d}\eta \\ &= m_1(T) + \int_0^\infty \left[a_\mathrm{F} \int_T^t \eta b(u) \times \mathrm{e}^{-\int_T^u \eta(p-q)b(\tau)\mathrm{d}\tau} \mathrm{d}u \right] f(\eta)\mathrm{d}\eta \quad t \ge T \\ &= m_1(T) + \int_T^t a_\mathrm{F} b(u) \left[\int_0^\infty \eta \times \mathrm{e}^{-\eta\int_T^u (p-q)b(\tau)\mathrm{d}\tau} f(\eta)\mathrm{d}\eta \right] \mathrm{d}u , \end{aligned} \qquad (27.7)$$

where a_F is the number of faults in the software at time T. Using the Laplace transform formula, the mean-value function can be rewritten as

$$\begin{aligned} m_2(t) &= m_1(T) + \int_T^t a_\mathrm{F} b(u) \times \left(-\frac{\mathrm{d}F^*(s)}{\mathrm{d}s}\bigg|_{s=\int_0^u (p-q)b(\tau)\mathrm{d}\tau} \right) \mathrm{d}u , \\ &\quad t \ge T \\ &= m_1(T) + \frac{a_\mathrm{F}}{(p-q)} \times \int_T^t \left\{ -\mathrm{d}F^* \left[(p-q) \int_T^u b(\tau)\mathrm{d}\tau \right] \right\} , \end{aligned}$$

where $F^*(s)$ is the Laplace transform of the PDF $f(x)$ and

$$\int_0^\infty x\,\mathrm{e}^{-x\cdot s} f(x)\,\mathrm{d}x = -\frac{\mathrm{d}F^*(s)}{\mathrm{d}s}$$

or, equivalently,

$$\begin{aligned} m_2(t) &= m_1(T) - \frac{a_\mathrm{F}}{(p-q)} \\ &\quad \times F^*\left[(p-q)\int_T^u b(\tau)\,\mathrm{d}\tau\right]\Bigg|_T^t, \quad t \ge T \\ &= m_1(T) + \frac{a_\mathrm{F}}{(p-q)} \\ &\quad \times \left\{F^*(0) - F^*\left[(p-q)\int_T^t b(\tau)\,\mathrm{d}\tau\right]\right\}. \end{aligned}$$

Notice that $F^*(0) = \int_0^\infty \mathrm{e}^{-0x} f(x)\,\mathrm{d}x = 1$, so

$$\begin{aligned} m_2(t) &= m_1(T) + \frac{a_\mathrm{F}}{(p-q)} \\ &\quad \times \left\{1 - F^*\left[(p-q)\int_T^t b(\tau)\,\mathrm{d}\tau\right]\right\} \quad t \ge T. \end{aligned}$$

The expected number of faults in the software at time T is given by

$$\begin{aligned} a_\mathrm{F} &= a_1(T) - p m_1(T) \\ &= a\left[1 - \int_0^t (p-q) b(u)\,\mathrm{e}^{-\int_0^u (p-q)\cdot b(\tau)\,\mathrm{d}\tau}\,\mathrm{d}u\right] \\ &= a\,\mathrm{e}^{-\int_0^t (p-q) b(\tau)\,\mathrm{d}\tau}. \end{aligned}$$

The generalized RFE model can be obtained as

$$m(t) = \begin{cases} \frac{a}{(p-q)}\left(1 - \mathrm{e}^{-(p-q)\int_0^u b(\tau)\,\mathrm{d}\tau}\right) & t \le T \\ \frac{a}{(p-q)}\left\{1 - \mathrm{e}^{-(p-q)\int_0^T b(\tau)\,\mathrm{d}\tau}\right. \\ \left.\times F^*\left[(p-q)\int_T^t b(\tau)\,\mathrm{d}\tau\right]\right\} & t \ge T. \end{cases} \tag{27.8}$$

The model in (27.8) is a generalized software reliability model subject to random field environments. The next section presents specific RFE models for the gamma and beta distributions of the random field environmental factor η.

27.3 RFE Software Reliability Models

Obviously, the environmental factor η must be nonnegative. Any suitable nonnegative distribution may be used to describe the uncertainty about η. In this section we present two RFE models. The first model is a γ-RFE model, based on the gamma distribution, which can be used to evaluate and predict software reliability in field environments where the software failure-detection rate can be either greater or less than the failure detection rate in the testing environment. The second model is a β-RFE model, based on the beta distribution, which can be used to predict software reliability in field environments where the software failure detection rate can only be less than the failure detection rate in the testing environment.

27.3.1 γ-RFE Model

In this model, we use the gamma distribution to describe the random environmental factor η. This model is called the γ-RFE model.

Assume that η follows a gamma distribution with a probability density function as follows

$$f_\gamma(\eta) = \frac{\theta^\gamma \eta^{\gamma-1}\,\mathrm{e}^{-\theta\cdot\eta}}{\Gamma(\gamma)}, \gamma, \theta > 0; \eta \ge 0. \tag{27.9}$$

The gamma distribution has sufficient flexibility and has desirable qualities with respect to computations [27.18]. Figure 27.2 shows an example of the gamma density

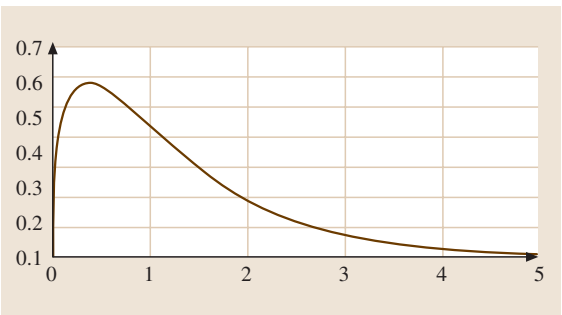


Fig. 27.2 A gamma density function

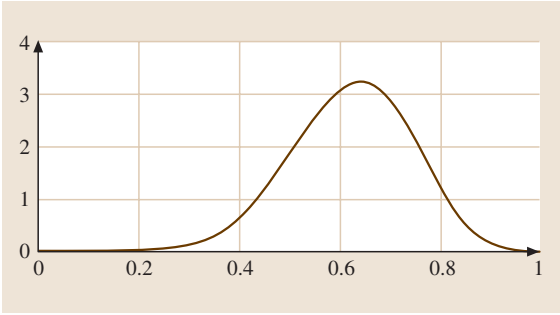


Fig. 27.3 A PDF curve of the beta distribution

probability function. The gamma function seems to be reasonable to describe a software failure process in those field environments where the software failure-detection rate can be either greater (i. e., $\eta > 1$) or less than (i. e., $\eta < 1$) the failure-detection rate in the testing environment.

The Laplace transform of the probability density function in (27.9) is

$$F^*(s) = \left(\frac{\theta}{\theta + s}\right)^{\gamma} . \tag{27.10}$$

Assume that the error-detection rate function $b(t)$ is given by

$$b(t) = \frac{b}{1 + c\,\mathrm{e}^{-b \cdot t}} , \tag{27.11}$$

where b is the asymptotic unit software-failure detection rate and c is the parameter defining the shape of the learning curve, then from (27.8) the mean-value function of the γ-RFE model can be obtained as follows

$$m_{\gamma}(t) = \begin{cases} \frac{a}{(p-q)}\left[1 - \left(\frac{1+c}{\mathrm{e}^{bt}+c}\right)^{(p-q)}\right] & t \le T , \\ \frac{a}{(p-q)}\left[1 - \left(\frac{1+c}{\mathrm{e}^{bT}+c}\right)^{(p-q)}\right. & \\ \left. \times \left(\frac{\theta}{\theta + (p-q)\ln\left(\frac{c+\mathrm{e}^{bt}}{c+\mathrm{e}^{bT}}\right)}\right)^{\gamma}\right] & t \ge T . \end{cases} \tag{27.12}$$

27.3.2 β-RFE Model

This section presents a model using the beta distribution that describes the random environmental factor η, called the β-RFE model.

The beta PDF is

$$f_{\beta}(\eta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\eta^{\alpha-1}(1-\eta)^{\beta-1} , \quad 0 \le \eta \le 1, \alpha > 0, \beta > 0 . \tag{27.13}$$

Figure 27.3 shows an example of the beta density function. It seems that the β-RFE model is a reasonable function to describe a software failure process in those field environments where the software failure-detection rate can only be less than the failure-detection rate in the testing environment. This is not uncommon in the software industry because, during software testing, the engineers generally test the software intensely and conduct an *accelerated* test on the software in order to detect most of the software faults as early as possible.

The Laplace transform of the PDF in (27.13) is

$$F^*_{\beta}(s) = \mathrm{e}^{-s} \cdot HG([\beta], [\alpha+\beta], s) , \tag{27.14}$$

where $HG([\beta], [\alpha+\beta], s)$ is a generalized hypergeometric function such that

$$HG([a_1, a_2, \ldots, a_m], [b_1, b_2, \ldots, b_n], s) = \sum_{k=0}^{\infty}\left(\frac{s^k \prod_{i=1}^{m}\frac{\Gamma(a_i+k)}{\Gamma(a_i)}}{\prod_{i=1}^{n}\frac{\Gamma(b_i+k)}{\Gamma(b_i)}k!}\right) ,$$

Therefore

$$\begin{aligned} F^*_{\beta}(s) &= \mathrm{e}^{-s}\sum_{k=0}^{\infty}\left(\frac{\Gamma(\alpha+\beta)\Gamma(\beta+k)s^k}{\Gamma(\beta)\Gamma(\alpha+\beta+k)k!}\right) \\ &= \sum_{k=0}^{\infty}\left(\frac{\Gamma(\alpha+\beta)\Gamma(\beta+k)}{\Gamma(\beta)\Gamma(\alpha+\beta+k)}\frac{s^k\,\mathrm{e}^{-s}}{k!}\right) \\ &= \sum_{k=0}^{\infty}\left(\frac{\Gamma(\alpha+\beta)\Gamma(\beta+k)}{\Gamma(\beta)\Gamma(\alpha+\beta+k)}\mathrm{Poisson}(k,s)\right) . \end{aligned}$$

where the Poisson probability density function is given by

$$\mathrm{Poisson}(k, s) = \frac{s^k\,\mathrm{e}^{-s}}{k!} .$$

Using the same error-detection rate function in (27.11) and replacing $F^*(s)$ by $F^*_{\beta}(s)$, the mean-value function of the β-RFE model is

$$m_{\beta}(t) = \begin{cases} \frac{a}{(p-q)}\left[1 - \left(\frac{1+c}{\mathrm{e}^{bt}+c}\right)^{(p-q)}\right] & t \le T , \\ \frac{a}{(p-q)}\left[1 - \left(\frac{1+c}{\mathrm{e}^{bT}+c}\right)^{(p-q)}\right. & \\ \left. \times \sum_{k=0}^{\infty}\left(\frac{\Gamma(\alpha+\beta)\Gamma(\beta+k)\mathrm{Poisson}(k,s)}{\Gamma(\beta)\Gamma(\alpha+\beta+k)}\right)\right] & t \ge T . \end{cases} \tag{27.15}$$

where

$$s = (p-q)\left[\ln\left(\frac{c+\mathrm{e}^{bt}}{c+\mathrm{e}^{bT}}\right)\right] .$$

27.4 Parameter Estimation

27.4.1 Maximum Likelihood Estimation (MLE)

We use the MLE method to estimate the parameters in these two RFE models. Let y_i be the cumulative number of software faults detected up to time t_i, $i = 1, 2, \ldots, n$. Based on the NHPP, the likelihood function is given by

$$L = \prod_{i=1}^{n} \frac{\left[m(t_i) - m(t_{i-1})\right]^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} \mathrm{e}^{-\left[m(t_i) - m(t_{i-1})\right]} . \tag{27.16}$$

The logarithmic form of the above likelihood function is

$$\ln L = \sum_{i=1}^{n} \left\{ (y_i - y_{i-1}) \ln \left[m(t_i) - m(t_{i-1})\right] - \left[m(t_i) - m(t_{i-1})\right] - \ln \left[(y_i - y_{i-1})!\right] \right\} . \tag{27.17}$$

In this analysis, the error-removal efficiency p is given. Each model has five unknown parameters. For example, in the γ-RFE model, we need to estimate the following five unknown parameters: a, b, q, γ and θ. For the β-RFE model, we need to estimate: a, b, q, α and β. By taking derivatives of (27.18) with respect to each parameter and setting the results equal to zero, we can obtain five equations for each RFE model. After solving all those equations, we obtain the maximum likelihood estimates (MLEs) of all parameters for each RFE model.

Table 27.2 Normalized cumulative failures and times during software testing

Time	Failures	Time	Failures	Time	Failures
0.0001	0.0249	0.0038	0.3483	0.0121	0.6766
0.0002	0.0299	0.0044	0.3532	0.0128	0.7015
0.0002	0.0647	0.0048	0.3682	0.0135	0.7363
0.0003	0.0647	0.0053	0.3881	0.0142	0.7761
0.0005	0.1095	0.0058	0.4478	0.0147	0.7761
0.0006	0.1194	0.0064	0.4876	0.0155	0.8159
0.0008	0.1443	0.0070	0.5224	0.0164	0.8259
0.0012	0.1692	0.0077	0.5473	0.0172	0.8408
0.0016	0.1990	0.0086	0.5821	0.0176	0.8458
0.0023	0.2289	0.0095	0.6119	0.0180	0.8756
0.0028	0.2637	0.0105	0.6368	0.0184	0.8955
0.0033	0.3134	0.0114	0.6468	0.0184	0.9005

Table 27.2 shows a set of failure data from a telecommunication software application during software testing [27.16]. The column "Time" shows the normalized cumulative time spent in software testing for this telecommunication application, and the column "Failures" shows the normalized cumulative number of failures occurring in the testing period up to the given time.

The time to stop testing is $T = 0.0184$. After the time T, the software is released for field operations. Table 27.3 shows the field data for this software release. Similarly, the column "Time" shows the normalized cumulative time spent in the field for this software application, and the time in Table 27.3 is continued from the time to stop testing T. The column "Failures" shows the normalized cumulative number of failures found after releasing the software for field operations up to the given time. The cumulative number of failures is the total number of software failures since the beginning of software testing.

To obtain a better understanding of the software development process, we show the actual results of the MLE solutions instead of the normalized results. In this study, let us assume that testing engineers have a number of years of experience of this particular product and software development skills and therefore conducted perfect debugging during the test. In other word, $p = 1$. The maximum likelihood estimates of all the parameters in the γ-RFE model are obtained as shown in Table 27.4.

Table 27.3 Normalized cumulative failures and their times in operation

Time	Failures	Time	Failures	Time	Failures
0.0431	0.9055	0.3157	0.9751	0.7519	0.9900
0.0616	0.9104	0.3407	0.9751	0.7585	0.9900
0.0801	0.9204	0.3469	0.9751	0.7718	0.9900
0.0863	0.9254	0.3967	0.9751	0.7983	0.9900
0.1357	0.9303	0.4030	0.9801	0.8251	0.9900
0.1419	0.9353	0.4291	0.9851	0.8453	0.9900
0.1666	0.9453	0.4357	0.9851	0.8520	0.9900
0.2098	0.9453	0.4749	0.9851	0.9058	0.9900
0.2223	0.9502	0.5011	0.9851	0.9126	0.9900
0.2534	0.9502	0.5338	0.9851	0.9193	0.9900
0.2597	0.9502	0.5731	0.9851	0.9395	0.9950
0.2659	0.9502	0.6258	0.9900	0.9462	0.9950
0.2721	0.9552	0.6656	0.9900	0.9529	1.0000
0.2971	0.9602	0.6789	0.9900	0.9865	1.0000
0.3033	0.9701	0.7253	0.9900	1.0000	1.0000

Table 27.4 MLE solutions for the γ-RFE model

$\hat{a}$	$\hat{b}$	$\hat{q}$	$\hat{c}$	$\hat{\gamma}$	$\hat{\theta}$
236.58	0.001443	0	0	0.2137	10.713

Table 27.5 MLE solutions for the β-RFE model

$\hat{a}$	$\hat{b}$	$\hat{q}$	$\hat{c}$	$\hat{\alpha}$	$\hat{\beta}$
236.07	0.001449	0	0	0.1862	8.6922

Similarly, set $p = 1$, the MLE of all the parameters in the β-RFE model are obtained as shown in Table 27.5.

For both RFE models, the MLE results can be used to obtain more insightful information about the software development process. In this example, at the time to stop testing the software $T = 0.0184$, the estimated number of remaining faults in the system is $a_F = a - (p-q)m(T) = 55$.

27.4.2 Mean-Value Function Fits

After we obtain the MLEs for all the parameters, we can plot the mean-value function curve fits for both the γ-RFE and β-RFE models based on the MLE parameters against the actual software application failures.

Table 27.6 shows the mean-value function curve fits for both the models where the column $m_\gamma(t)$ and $m_\beta(t)$ show the mean-value function for the γ-RFE model and the β-RFE model, respectively.

The γ-RFE and β-RFE models yield very close fits and predictions on software failures. Figure 27.4 shows the mean-value function curve fits for both the γ-RFE model and β-RFE model. Both models appear to be a good fit for the given data set. Since we are particularly interested in the fits and the predictions for software failure data during field operation, we also plot the detailed mean-value curve fits for both the γ-RFE model and the β-RFE model in Fig. 27.5.

For the overall fitting of the mean-value function against the actual software failures, the mean squared error (MSE) is 23.63 for the γ-RFE model fit, and is 23.69 for the β-RFE model. We can also obtain the fits and predictions for software failures by applying some existing NHPP software reliability models to the same set of failure data. Since all these existing models assumes a constant failure-detection rate throughout both the software testing and operation periods, we only apply the software testing data to the software models and then predict the software failures in the field environments.

Figure 27.6 shows the comparisons of the mean-value function curve fits between the two RFE models and some existing NHPP software reliability models. It appears that the two models that

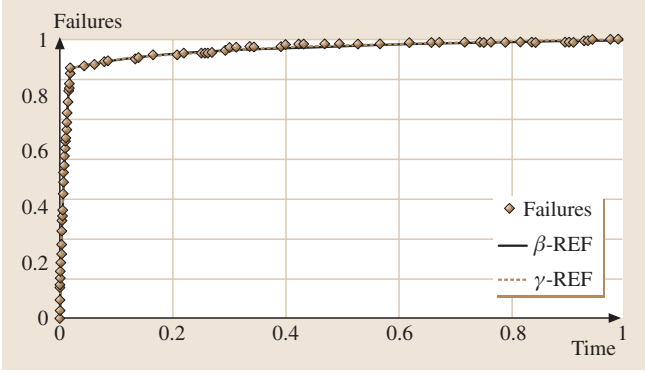


Fig. 27.4 Mean-value function curve fits for both RFE models

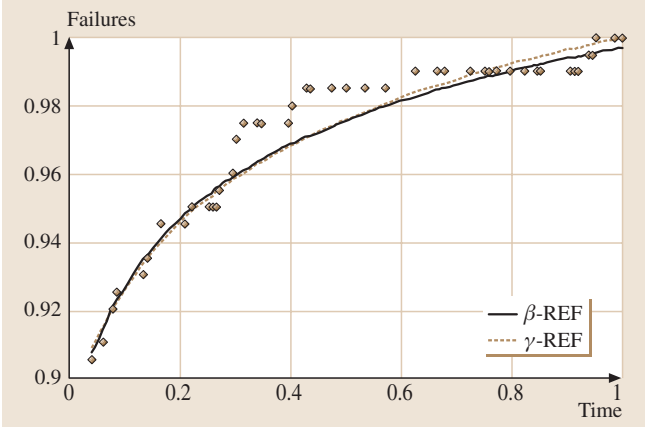


Fig. 27.5 Mean-value function fitting comparisons

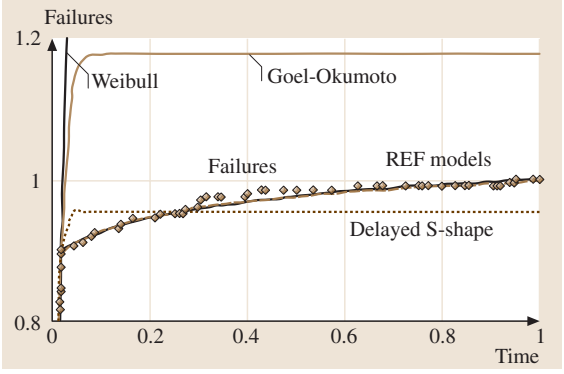


Fig. 27.6 Model comparisons

Table 27.6 The mean-value functions for both RFEs models

Time	Failures	$m_\gamma(t)$	$m_\beta(t)$	Time	Failures	$m_\gamma(t)$	$m_\beta(t)$
0.0000	0.0000	0.0000	0.0000	0.1357	0.9303	0.9340	0.9341
0.0001	0.0249	0.0085	0.0085	0.1419	0.9353	0.9352	0.9354
0.0002	0.0299	0.0152	0.0152	0.1666	0.9453	0.9398	0.9399
0.0002	0.0647	0.0219	0.0219	0.2098	0.9453	0.9469	0.9467
0.0003	0.0647	0.0302	0.0302	0.2223	0.9502	0.9487	0.9485
0.0005	0.1095	0.0466	0.0467	0.2534	0.9502	0.9530	0.9525
0.0006	0.1194	0.0547	0.0548	0.2597	0.9502	0.9538	0.9533
0.0008	0.1443	0.0708	0.0709	0.2659	0.9502	0.9545	0.9540
0.0012	0.1692	0.1023	0.1025	0.2721	0.9552	0.9553	0.9547
0.0016	0.1990	0.1404	0.1406	0.2971	0.9602	0.9582	0.9575
0.0023	0.2289	0.1915	0.1917	0.3033	0.9701	0.9589	0.9582
0.0028	0.2637	0.2332	0.2335	0.3157	0.9751	0.9603	0.9594
0.0033	0.3134	0.2667	0.2670	0.3407	0.9751	0.9628	0.9618
0.0038	0.3483	0.3053	0.3056	0.3469	0.9751	0.9635	0.9624
0.0044	0.3532	0.3422	0.3426	0.3967	0.9751	0.9681	0.9667
0.0048	0.3682	0.3718	0.3721	0.4030	0.9801	0.9686	0.9672
0.0053	0.3881	0.4003	0.4007	0.4291	0.9851	0.9708	0.9692
0.0058	0.4478	0.4332	0.4336	0.4357	0.9851	0.9713	0.9697
0.0064	0.4876	0.4648	0.4651	0.4749	0.9851	0.9743	0.9725
0.0070	0.5224	0.4998	0.5002	0.5011	0.9851	0.9761	0.9742
0.0077	0.5473	0.5332	0.5335	0.5338	0.9851	0.9783	0.9762
0.0086	0.5821	0.5772	0.5775	0.5731	0.9851	0.9808	0.9785
0.0095	0.6119	0.6205	0.6208	0.6258	0.9900	0.9839	0.9813
0.0105	0.6368	0.6600	0.6602	0.6656	0.9900	0.9860	0.9833
0.0114	0.6468	0.6953	0.6955	0.6789	0.9900	0.9867	0.9839
0.0121	0.6766	0.7210	0.7211	0.7253	0.9900	0.9890	0.9860
0.0128	0.7015	0.7479	0.7479	0.7519	0.9900	0.9902	0.9871
0.0135	0.7363	0.7684	0.7684	0.7585	0.9900	0.9905	0.9874
0.0142	0.7761	0.7924	0.7924	0.7718	0.9900	0.9911	0.9879
0.0147	0.7761	0.8050	0.8049	0.7983	0.9900	0.9923	0.9890
0.0155	0.8159	0.8294	0.8292	0.8251	0.9900	0.9934	0.9900
0.0164	0.8259	0.8522	0.8520	0.8453	0.9900	0.9943	0.9908
0.0172	0.8408	0.8713	0.8710	0.8520	0.9900	0.9945	0.9910
0.0176	0.8458	0.8804	0.8801	0.9058	0.9900	0.9966	0.9929
0.0180	0.8756	0.8897	0.8893	0.9126	0.9900	0.9969	0.9932
0.0184	0.8955	0.8987	0.8983	0.9193	0.9900	0.9971	0.9934
0.0184	0.9005	0.8995	0.8991	0.9395	0.9950	0.9979	0.9941
0.0431	0.9055	0.9092	0.9092	0.9462	0.9950	0.9981	0.9943
0.0616	0.9104	0.9153	0.9155	0.9529	1.0000	0.9983	0.9945
0.0801	0.9204	0.9208	0.9210	0.9865	1.0000	0.9995	0.9956
0.0863	0.9254	0.9224	0.9227	1.0000	1.0000	1.0000	0.9960

include consideration of the field environment on the software failure-detection rate perform better in terms of the predictions for software failures in the field.

27.4.3 Software Reliability

Once the MLEs of all the parameters in (27.12) and (27.14) are obtained, the software reliability within

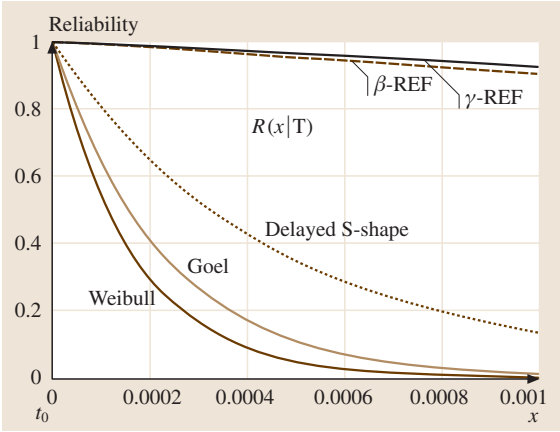


Fig. 27.7 Reliability prediction comparisons

$(t, t+x)$ can be determined as

$$R(x|t) = \mathrm{e}^{-[m(t+x)-m(t)]} \,. \tag{27.18}$$

Let $T = 0.0184$, and change x from 0 to 0.004, then we can compare the reliability predictions between the two RFE models and some other NHPP models that assume a constant failure-detection rate for both software testing and operation. The reliability prediction curves are shown in Fig. 27.7. From Fig. 27.7, we can see that the NHPP models without consideration of the environmental factor yield much lower predictions for software reliability in the field than the two proposed RFE software reliability models.

27.4.4 Confidence Interval

γ-RFE Model

To see how good the reliability predictions given by the two RFE models are, in this section we describe how to construct confidence intervals for the prediction of software reliability in the random field environments. From Tables 27.4 and 27.5, the MLEs of c and q are equal to zero and, if p is set to 1, then the model in (27.12) becomes

$$m(t) = \begin{cases} a\left(1 - \mathrm{e}^{-b\cdot t}\right) & t \le T \,, \\ a\left[1 - \mathrm{e}^{-b\cdot T}\left(\frac{\theta}{\theta + b(t-T)}\right)^{\gamma}\right] & t \ge T \,. \end{cases} \tag{27.19}$$

This model leads to the same MLE results for the parameters a, b, γ and θ, and also yields exactly the same mean-value function fits and predictions as the model in (27.12). To obtain the confidence interval for the reliability predictions for the γ-RFE model, we derive the variance–covariance matrix for all the maximum likelihood estimates as follows.

If we use x_i, $i = 1, 2, 3$, and 4, to denote all the parameters in the model, or

$$x_1 \to a \quad x_2 \to b \quad x_3 \to \theta \quad x_4 \to \gamma \,.$$

The Fisher information matrix H can be obtained as

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix} \,, \tag{27.20}$$

where

$$h_{ij} = E\left(-\frac{\partial^2 L}{\partial x_i \partial x_j}\right) i, j = 1, \cdots, 6 \,, \tag{27.21}$$

where L is the log-likelihood function in (27.18).

If we denote $z(t_k) = m(t_k) - m(t_{k-1})$ and $\Delta y_k = y_k - y_{k-1}$, $k = 1, 2, \ldots, n$, then we have

$$\frac{\partial^2 L}{\partial x_i \partial x_j} = \sum_{k=1}^{n}\left[-\frac{\Delta y_k}{z(t_k)^2}\frac{\partial z(t_k)}{\partial x_i}\cdot\frac{\partial z(t_k)}{\partial x_j} + \left(\frac{\Delta y_k - z(t_k)}{z(t_k)}\cdot\frac{\partial^2 z(t_k)}{\partial x_i \partial x_j}\right)\right] \,. \tag{27.22}$$

Then we can obtain each element in the Fisher information matrix H. For example,

$$\begin{aligned} h_{11} &= E\left(-\frac{\partial^2 L}{\partial x_1^2}\right) \\ &= \sum_{k=1}^{n}\left\{\sum_{\Delta y_k=0}^{\infty}\left[\frac{\Delta y_k}{z(t_k)^2}\left(\frac{\partial z(t_k)}{\partial a}\right)^2\right] \times \frac{[z(t_k)]^{\Delta y_k}\,\mathrm{e}^{-z(t_k)}}{(\Delta y_k)!}\right\} \\ &= \sum_{k=1}^{n}\left\{\sum_{\Delta y_k=0}^{\infty}\left[\frac{\Delta y_k}{z(t_k)^2}\left(\frac{z(t_k)}{a}\right)^2\right] \times \frac{[z(t_k)]^{\Delta y_k}\,\mathrm{e}^{-z(t_k)}}{(\Delta y_k)!}\right\} \\ &= \sum_{k=1}^{n}\left(\frac{1}{a^2}\sum_{\Delta y_k=0}^{\infty}\Delta y_k \frac{[z(t_k)]^{\Delta y_k}\,\mathrm{e}^{-z(t_k)}}{(\Delta y_k)!}\right) \\ &= \sum_{k=1}^{n}\left[\frac{1}{a^2}\cdot z(t_k)\right] \\ &= \frac{1}{a^2}m(t_n) \,. \end{aligned} \tag{27.23}$$

The variance matrix, V, can also be obtained

$$V = (H)^{-1} = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix} . \tag{27.24}$$

The variances of all the estimate parameters are given by

$$\begin{aligned} \mathrm{Var}(\hat{a}) &= \mathrm{Var}(x_1) = v_{11} , \\ \mathrm{Var}(\hat{b}) &= \mathrm{Var}(x_2) = v_{22} , \\ \mathrm{Var}(\hat{\gamma}) &= \mathrm{Var}(x_3) = v_{33} , \\ \mathrm{Var}(\hat{\theta}) &= \mathrm{Var}(x_4) = v_{44} . \end{aligned} \tag{27.25}$$

The actual numerical results for the γ-RFE model variance matrix are

$$V_\gamma = \begin{pmatrix} 703.8472 & -0.005387 & -88.6906 & -2.6861 \\ -0.005387 & 7.3655\times10^{-8} & 1.11\times10^{-3} & 3.097\times10^{-5} \\ -88.6906 & 1.11\times10^{-3} & 92.4287 & 1.1843 \\ -2.6861 & 3.097\times10^{-5} & 1.1843 & 0.0238 \end{pmatrix} . \tag{27.26}$$

β-RFE Model

The model in (27.14) can also be simplified given that the estimates of both q and c are equal to zero and p is set to 1. The mean-value function becomes

$$m_\beta(t) = \begin{cases} a(1-\mathrm{e}^{-bt}) & t \le T , \\ a\left[1-\mathrm{e}^{-bT} \right. & \\ \left. \times \sum_{k=0}^{\infty} \left(\frac{\Gamma(\alpha+\beta)\Gamma(\beta+k)\mathrm{Poisson}[k,b(t-T)]}{\Gamma(\beta)\Gamma(\alpha+\beta+k)} \right) \right] & \\ & t \ge T . \end{cases} \tag{27.27}$$

This model leads to the same MLE results for the parameters a, b, α and β, and also yields exactly the same mean-value function fits and predictions. To obtain the confidence interval for the reliability predictions for the β-RFE model, we need to obtain the variance–covariance matrix for all the maximum likelihood estimates.

If we use x_i, $i = 1, 2, 3$, and 4, to denote all the parameters in the model, or

$$x_1 \to a \quad x_2 \to b \quad x_3 \to \alpha \quad x_4 \to \beta ,$$

and go through similar steps as for the γ-RFE model, the actual numerical results for the β-RFE model variance matrix can be obtained as

$$V_\beta = \begin{pmatrix} 691.2 & -0.00536 & -2.728 & -66.2172 \\ -0.00536 & 7.4485\times10^{-8} & 2.671\times10^{-5} & 0.00085 \\ -2.7652 & 2.671\times10^{-5} & 0.01820 & 0.8295 \\ -66.2172 & 0.00085 & 0.8295 & 60.5985 \end{pmatrix} . \tag{27.28}$$

Confidence Interval of the Reliability Predictions

If we define a partial derivative vector for the reliability $R(x|t)$ in (27.18) as

$$vR(x|t) = \left(\frac{\partial R(x|t)}{\partial x_1}, \frac{\partial R(x|t)}{\partial x_2}, \frac{\partial Rb(x|t)}{\partial x_3}, \frac{\partial R(x|t)}{\partial x_4} \right) \tag{27.29}$$

then the variance of $R(x|t)$ in (27.18) can be obtained as

$$\mathrm{Var}\left[R(x|t)\right] = vR(x|t)V\left[vR(x|t)\right]^T . \tag{27.30}$$

Assume that the reliability estimation follows a normal distribution, then the 95% confidence interval for the reliability prediction $R(x|t)$ is

$$\begin{aligned} &\left[R(x|t) - 1.96\times\sqrt{\mathrm{Var}\left[R(x|t)\right]} \right. , \\ &\left. R(x|t) + 1.96\times\sqrt{\mathrm{Var}\left[R(x|t)\right]} \right] . \end{aligned} \tag{27.31}$$

Figures 27.8 and 27.9 show the 95% confidence interval of the reliability predicted by the γ-RFE and β-RFE models, respectively.

We plot the reliability predictions and their 95% confidence interval for both the γ-RFE model and the

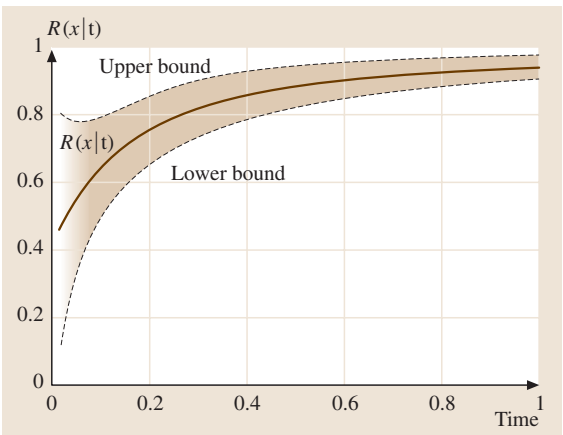


Fig. 27.8 γ-RFE model reliability growth curve and its 95% confidence interval

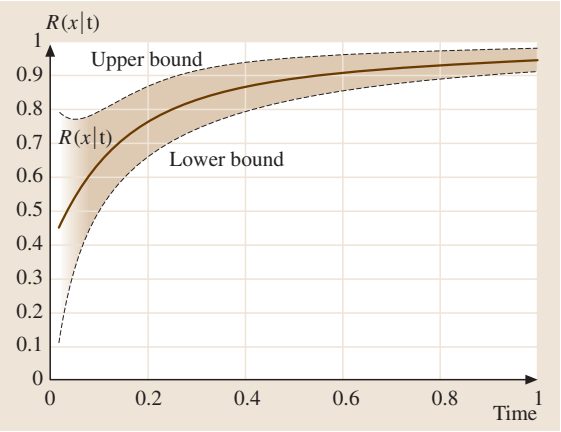


Fig. 27.9 β-RFE model reliability growth prediction and its 95% confidence interval

β-RFE model in Fig. 27.10. For this given application data set, the reliability predictions for the γ-RFE model and the β-RFE model are very close to each other, as are their confidence intervals. Therefore, it would not matter too much which one of the two RFE models were used to evaluate the software reliability for this application. However, will these two RFE models always yield similar reliability predictions for all software applications? or, which model should one choose for applications if they are not always that close to each other? We will try to answer these two questions in the next section. Figure 27.11 shows the 95% confidence interval for the mean-value function fits and predictions from the γ-RFE model.

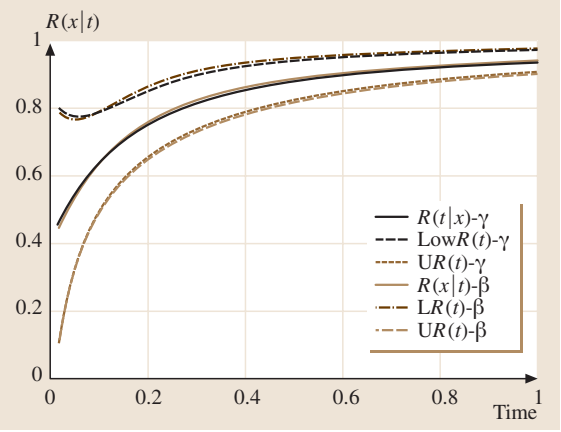


Fig. 27.10 Reliability growth prediction curves and their 95% confidence intervals for the γ-RFE model and the β-RFE model

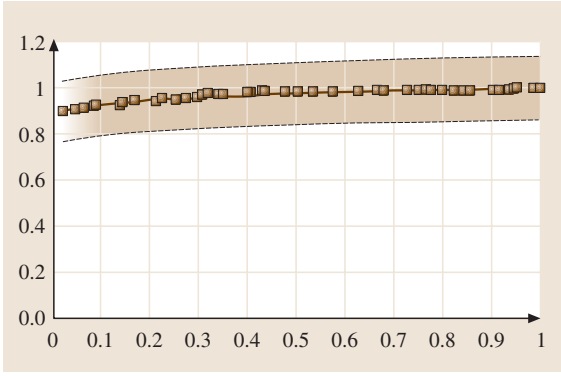


Fig. 27.11 Mean-value function curve fit and its 95% confidence intervals for the γ-RFE model

27.4.5 Concluding and Remarks

Table 27.7 shows all the maximum likelihood estimates of all the parameters and other fitness measures. The maximum likelihood estimates (MLEs) on common parameters, such as a—the initial number of faults in the software, and b—the unit software failure-detection rate during testing, are consistent for both models. Both models provide very close predictions for software reliability and also give similar results for the mean and variance of the random environment factor η.

The underlying rationale for this phenomenon is the similarity between the gamma and beta distributions when the random variable η is close to zero. In this application, the field environments are much less liable to software failure than the testing environment. The random field environmental factor, η, is mostly much less than 1 with mean $(\eta) \approx 0.02$.

Figure 27.12 shows the probability density function curves of the environmental factor η based on the MLEs of all the parameters for both the γ-RFE model and

Table 27.7 MLEs and fitness comparisons

Parameter	γ-RFE	β-RFE
$\hat{a}$	236.5793016	236.0745369
$\hat{b}$	0.001443362	0.001448854
$\hat{\theta}$	10.7160153	
$\hat{\gamma}$	0.213762945	
$\hat{\alpha}$		0.186224489
$\hat{\beta}$		8.692191792
Mean	0.019948	0.020975
Variance	0.0018615	0.002079
MSE	23.63	23.69
Likelihood	**−136.1039497**	**−129.7811199**

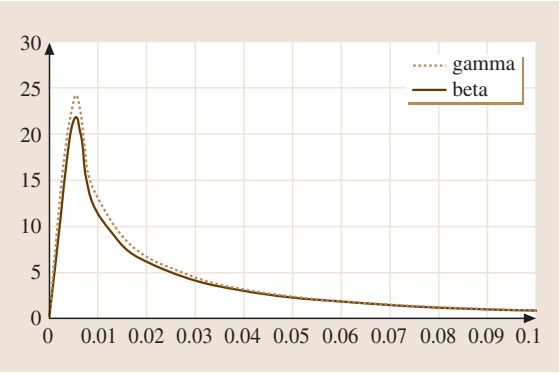


Fig. 27.12 Probability density function curves comparison for the environmental factor η

the β-RFE model. We observe that the PDF curves for the beta and gamma distributions are also very close to each other. The two RFEs models give similar results because this software application is much less likely to fail in the field environment, with mean $(\eta) = 0.02$. If the mean (η) is not so close to 0, then we would expect to have different prediction results from the γ-RFE model and the β-RFE model.

We suggest the following criteria as ways to select between the two models discussed in this chapter for predicting the software reliability in the random field environments:

1. Software less liable to failure in the field than in testing, i. e., $\eta \leq 1$

   In the γ-RFE model, the random field environmental factor, η following a gamma distribution, ranges from 0 to $+\infty$. For the β-RFE model, the random field environmental factor, η following a beta distribution, ranging from 0 to 1. Therefore, the β-RFE model will be more appropriate for describing field environments in which the software application is likely to fail than in the controlled testing environment.

   For this given application, we notice that, when the field environmental factor η is much less than 1 [mean$(\eta) = 0.02$], the γ-RFE model yields similar results to the β-RFE model. However, we also observe that the γ-RFE model does not always yield similar results to the β-RFE model when η is not close to 0. In this case, if we keep using the γ-RFE model instead of the β-RFE model, we would expect to see a large variance in the maximum likelihood estimates for all the unknown parameters, and hence a wider confidence interval for the reliability prediction.
2. Smaller variance of the RFE factor η

   A smaller variance of the random environmental factor η will generally lead to a smaller confidence interval for the software reliability prediction. It therefore represents a better prediction in the random field environments.
3. Smaller variances for the common parameters a and b

   The software parameter a and the process parameter b are directly related to the accuracy of reliability prediction. They can also be used to investigate the software development process. Smaller variances of a and b would lead, in general, to smaller confidence intervals for the mean-value function predictions and reliability predictions.
4. Smaller mean squared error (MSE) of the mean-value function fits

   A smaller MSE for the mean-value function fits means a better fit of the model to the real system failures. This smaller MSE will usually lead to a better prediction of software failures in random field environments.

The above criteria can be used with care to determine which RFE model should be chosen in practice. They may sometime provide contradictory results. In the case of contradictions, practitioners can often consider selecting the model with the smaller confidence interval for the reliability prediction.

References

27.1 H. Pham, X. Zhang: A software cost model with warranty and risk costs, IEEE Trans. Comput. **48**, 71–75 (1999)

27.2 H. Pham, L. Normann, X. Zhang: A general imperfect debugging NHPP model with S-shaped fault detection rate, IEEE Trans. Reliab. **48**, 169–175 (1999)

27.3 A. L. Goel, K. Okumoto: Time-dependent error-detection rate model for software and other performance measures, IEEE Trans. Reliab. **28**, 206–211 (1979)

27.4 M. Ohba: Software reliability analysis models, IBM J. Res. Dev. **28**, 428–443 (1984)

27.5 H. Pham: *Software Reliability* (Springer, London 2000)

27.6 S. Yamada, M. Ohba, S. Osaki: S-shaped reliability growth modeling for software error detection, IEEE Trans. Reliab. **33**, 475–484 (1984)

27.7 X. Zhang, X. Teng, H. Pham: Considering fault removal efficiency in software reliability assessment, IEEE Trans. Syst. Man Cybern. A **33**, 114–120 (2003)
27.8 H. Pham, X. Zhang: NHPP software reliability and cost models with testing coverage, Eur. J. Oper. Res. **145**, 443–454 (2003)
27.9 X. Zhang, H. Pham: Predicting operational software availability and its Applications to telecommunication systems, Int. J. Syst. Sci. **33**(11), 923–930 (2002)
27.10 H. Pham, H. Wang: A quasi renewal process for software reliability and testing costs, IEEE Trans. Syst. Man Cybern. A **31**, 623–631 (2001)
27.11 X. Zhang, Mi-Young Shin: Exploratory analysis of environmental factors for enhancing the software reliability assessment, J. Syst. Softw. **57**, 73–78 (2001)
27.12 L. Pham, H. Pham: A Bayesian predictive software reliability model with pseudo-failures, IEEE Trans. Syst. Man Cybern. A **31**(3), 233–238 (2001)
27.13 X. Zhang, H. Pham: Comparisons of nonhomogeneous Poisson process software reliability models and its applications, Int. J. Syst. Sci. **31**(9), 1115–1123 (2000)
27.14 H. Pham: Software reliability and cost models: perspectives, comparison and practice, Eur. J. Oper. Res. **149**, 475–489 (2003)
27.15 B. Yang, M. Xie: A study of operational, testing reliability in software reliability analysis, Reliab. Eng. Syst. Safety **70**, 323–329 (2000)
27.16 X. Zhang, D. Jeske, H. Pham: Calibrating software reliability models when the test environment does not match the user environment, Appl. Stochastic Models Bus. Ind. **18**, 87–99 (2002)
27.17 D. R. Cox: Regression models and life tables (with discussion), J. R. Stat. Soc. Ser. B **34**, 133–144 (1972)
27.18 X. Teng, H. Pham: A software cost model for quantifying the gain with considerations of random field environment, IEEE Trans. Comput. **53**, 3 (2004)

Part D Regression

Part D Regression Methods and Data Mining

Part D focuses on regression methods and data mining. The first chapter in this part, Chapt. 28, describes various diagnostic procedures for detecting single and multiple outliers and influential observations in linear regression. It also discusses procedures for detecting high-leverage outliers in large, high-dimensional data sets. Chapter 29 gives an overview of various logistic regression methods for fitting models to a binary-valued response variable and introduces the idea of a logistic regression tree based on a recursive partitioning algorithm to fit a linear logistic regression model for solving large, complex data sets. Chapter 30 introduces the basic structure of tree-based methods for constructing trees for both classification and regression problems by recursively partitioning a learning sample over its input variable space. It also compares classification and regression trees to multivariate adaptive regression splines, neural networks and support-vector machines. Chapter 31 presents the concept of a generalization of least-squares estimation (LSE), called M estimators, to solve the statistical problems involving unknown coordinate systems and image registration problems. This chapter also discusses in detail the differences between the LSE and M estimators and presents the statistical properties of M estimates for spherical regression.

The following three chapters focus on the statistical analysis of genomic and proteomics data. Chapter 32 provides an overview of the emerging statistical concepts of statistical genetics, which are commonly used to analyze microarray gene-expression data, and further introduces recent statistical testing methods, such as significance analysis of microarray and local pooled-error tests, as well as supervised-learning discovery tools. Chapter 33 describe several statistical methods, such as the empirical Bayesian approach, significance analysis of microarray, support-vector machines, and tree- and forest-based classification, for analyzing genomic data and their applications in biochemical and genetic research. Chapter 34 discusses two proteomics statistical techniques, disease biomarker discovery and protein/peptide identification, and their applications in both the biological and medical research for analyzing mass-spectrometry data. The next two chapters focus on data mining and its applications. Chapter 35 describes the radical basis-function model architecture and its applications in bio-informatics and biomedical engineering and also describes the four algorithms commonly used for its design: clustering, orthogonal least squares, regularization, and gradient descent, while Chapt. 36 presents the basic principles of data-mining methodologies in databases, including knowledge discovery, supervised learning, software, the classification problem, neural networks, and association rules, and discusses several popular data-mining methods with applications in industry and business practice.

28. Measures of Influence and Sensitivity in Linear Regression

This chapter reviews diagnostic procedures for detecting outliers and influential observations in linear regression. First, the statistics for detecting single outliers and influential observations are presented, and their limitations for multiple outliers in high-leverage situations are discussed; second, diagnostic procedures designed to avoid masking are shown. We comment on the procedures by *Hadi* and *Smirnoff* [28.1,2], *Atkinson* [28.3] and *Swallow* and *Kianifard* [28.4] based on finding a clean subset for estimating the parameters and then increasing its size by incorporating new homogeneous observations one by one, until a heterogeneous observation is found. We also discuss procedures for detecting high-leverage outliers in large data sets based on eigenvalue analysis of the influence and sensitivity matrix, as proposed by *Peña* and *Yohai* [28.5, 6]. Finally we show that the joint use of simple univariate statistics, as predictive residuals, and Cook's distances, jointly with the sensitivity statistic proposed by *Peña* [28.7] can be a useful diagnostic tool for large high-dimensional data sets.

Data often contain outliers or atypical observations. Outliers are observations which are heterogeneous with the rest of the data, due to large measurement errors, different experimental conditions or unexpected effects. Detecting these observations is important because they can lead to new discoveries. For instance, penicillin was found because Pasteur, instead of ignoring an outlier, tried to understand the reason for this atypical effect. As *Box* [28.8] has emphasized "every operating system supplies information on how it can be improved and if we use this information it can be a source of continuous improvement". A way in which this information appears is by outlying observations, but in many engineering processes these observations are not easy to detect. For instance, in a production process a large value in one of the variables we monitor may be due, among other causes, to: (1) a large value of one of the input control variables; (2) an unexpected interaction among the input variables; (3) a large measurement error due to some defect in the measurement instrument. In the first case, the outlying observations may provide no new information about the performance of the process but in the second case may lead to a potentially useful discovery and in the third, to an improvement of the process control. A related problem is to avoid the situation where these outliers affect the estimation of the statistical model and this is the aim of robust estimation methods.

This chapter discusses outliers, influential and sensitive observations in regression models and presents methods to detect them. Influential observations are those which have a strong influence on the global properties of the model. They are obtained by modifying the weights attached to each case, and looking at the standardized change of the parameter vector or the vector of forecasts. Influence is a global analysis. Sensitive observations can be declared outliers or not by small modifications in the sample. Sensitivity is more a local concept. We delete each sample point in turn and look at the change that these modifications produce in the forecast of a single point. We will see that influence

and sensitivity are important concepts for understanding the effect of data in building a regression model and in finding groups of outliers.

Many procedures are available to identify a single outlier or an isolated influential point in linear regression. The books of *Belsley* et al. [28.9], *Hawkins* [28.10], *Cook* and *Weisberg* [28.11], *Atkinson* [28.12], *Chatterjee* and *Hadi* [28.13] *Barnett* and *Lewis* [28.14] and *Atkinson* and *Riani* [28.15] present good analyses of this problem. To identify outliers and to measure influence the point can be deleted, as proposed by *Cook* [28.16] and *Belsley* et al. [28.9], or its weight decreased, as in the local influence analysis introduced by *Cook* [28.17]. See *Brown* and *Lawrence* [28.18] and *Suárez Rancel* and *González Sierra* [28.19] for a review of local influence in regression and many references, and *Hartless* et al. [28.20] for recent results on this approach. A related way to analyze influence has been proposed by *Critchley* et al. [28.21] by an extension of the influence-curve methodology. The detection of influential subsets or multiple outliers is more difficult, due to the masking and swamping problems. Masking occurs when one outlier is not detected because of the presence of others; swamping happens when a non-outlier is wrongly identified due to the effect of some hidden outliers, see *Lawrance* [28.22]. Several procedures have been proposed for dealing with multiple outliers, see *Hawkins*, *Bradu* and *Kass* [28.23], *Gray* and *Ling* [28.24], *Marasinghe* [28.25], *Kianifard* and *Swallow* [28.26, 27], *Hadi* and *Simonoff* [28.1, 2], *Atkinson* [28.3, 28] and *Swallow* and *Kianifard* [28.4]. A different analysis for detecting groups of outliers by looking at the eigenvectors of an influence matrix was presented by *Peña* and *Yohai* [28.5]. These authors later proposed [28.6] the sensitivity matrix as a better way to find interesting groups of data, and from this approach *Peña* [28.7] has proposed a powerful diagnostic statistic for detecting groups of outliers.

We do not discuss in this chapter, due to lack of space, robust regression methods and only refer to them when they are used as a first step in a diagnosis procedure. See *Huber* [28.29] for a good discussion of the complementary role of diagnosis and robustness. For robust estimation in regression see *Rousseeuw* and *Leroy* [28.30] and *Maronna*, *Martin* and *Yohai* [28.31]. Robust estimation of regression models has also received attention in the Bayesian literature since the seminal article of *Box* and *Tiao* [28.32]. See *Justel* and *Peña* [28.33] for a Bayesian approach to this problem and references.

The paper is organized as follows. In Sect. 28.1 we present the model and the notation, and define the main measures which will be used for outlier and influence analysis. In Sect. 28.2 we review procedures for detecting single outliers and influential observations in regression. In Sect. 28.3 we discuss the multiple-outlier problem and two types of diagnostic procedures, those based on an initial clean subset and those based on eigenvalue analysis of some diagnostic matrices. In Sect. 28.4 we introduce a simple statistic which can be used for diagnostic analysis of a large data set, avoiding the masking problem. Section 28.5 includes an example of the use of diagnostic methods for detecting groups of outliers and Sect. 28.6 contains some concluding remarks.

28.1 The Leverage and Residuals in the Regression Model

We assume that we have observed a sample of size n of a random variable $\boldsymbol{y} = (y_1, \ldots, y_n)'$ and a set of $p-1$ explanatory variables which are linearly related by

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + u_i, \tag{28.1}$$

where the u_i are the measurement errors, which will be independent normal zero-mean random variables with variance σ^2, and $\boldsymbol{u} = (u_1, \ldots, u_n)'$. The $\boldsymbol{x}_i = (1, x_{2i}, \ldots, x_{pi})$ are numerical vectors in R^p and we will denote by $\mathbf{X}$ the $n \times p$ matrix of rank p whose i-th row is $\boldsymbol{x}_i'$. Then, the least-squares estimate of $\boldsymbol{\beta}$ is obtained by projecting the vector $\boldsymbol{y}$ onto the space generated by the columns of $\mathbf{X}$, which leads to

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{y}$$

and the vector of fitted values, $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_n)'$, is given by

$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{H}\boldsymbol{y}, \tag{28.2}$$

where

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$$

is the idempotent projection matrix. The vector orthogonal to the space generated by the $\mathbf{X}$ variables is the residual vector, $\boldsymbol{e} = (e_1, \ldots, e_n)'$, which is defined

by

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} = (\mathbf{I} - \mathbf{H})\boldsymbol{y} \tag{28.3}$$

and we will let $\widehat{s}^2_{\mathrm{R}} = \boldsymbol{e}'\boldsymbol{e}/(n-p)$ be the estimated residual variance.

From (28.3), inserting $\mathbf{X}\boldsymbol{\beta} + \boldsymbol{u}$ instead of $\boldsymbol{y}$ and using $\mathbf{HX} = \mathbf{X}$, we obtain the relationship between the residuals and the measurement errors, $\boldsymbol{e} = (\mathbf{I} - \mathbf{H})\boldsymbol{u}$. Thus, each residual is a linear combination of the measurement errors. Letting $h_{ij} = \boldsymbol{x}_i' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \boldsymbol{x}_j$ be the elements of the matrix, $\mathbf{H}$, we have

$$e_i = u_i - \sum_{j=1}^{n} h_{ij} u_j \tag{28.4}$$

and, if the second term is small, the residual e_i will be close to the measurement error, u_i. The variance of this second term is

$$Var(\sum_{j=1}^{n} h_{ij} u_j) = \sigma^2 \sum_{j=1}^{n} h_{ij}^2 = \sigma^2 h_{ii}$$

and if h_{ii}, the diagonal term of $\mathbf{H}$, is large, the difference between the residual and the measurement error can be large. The values h_{ii} are called the leverage of the observation and measure the discrepancy of each observation $\boldsymbol{x}_i$ with respect to the mean of the explanatory variables. It can be shown (see for instance [28.11] p. 12) that

$$h_{ii} = \boldsymbol{x}_i'(\mathbf{X}'\mathbf{X})^{-1}\boldsymbol{x}_i = \frac{1}{n}\left[1 + \left(\widetilde{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}\right)' \mathbf{S}_{xx}^{-1} (\widetilde{\boldsymbol{x}}_i - \bar{\boldsymbol{x}})\right],$$

where $\widetilde{x}_h = (x_{2h}, \ldots, x_{ph})$ does not include the constant term, $\bar{\boldsymbol{x}}$ is the vector of means of the $p-1$ explanatory variables and $\mathbf{S}_{xx}$ is their covariance matrix. Note that, if the variables were uncorrelated, h_{ii} would be the sum of the standardized distances $[(x_{ij} - x_j)/s_j]^2$. As $\sum_{i=1}^{n} h_{ii} = \mathrm{tr}(\mathbf{H}) = p$, the average value of the leverage is $\bar{h} = \sum h_{ii}/n = p/n$, and it can be shown that $1/n \le h_{ii} \le 1$. From (28.4) we conclude that the residual will be close to the measurement error for those observations close to the center of the explanatory data, where $h_{ii} \simeq 1/n$, but will be very different for the extreme points where $h_{ii} \simeq 1$. Note that the residual covariance matrix is

$$\begin{aligned} Var(\boldsymbol{e}) = E[\boldsymbol{e}\boldsymbol{e}'] &= E[(\mathbf{I} - \mathbf{H})\boldsymbol{u}\boldsymbol{u}'(\mathbf{I} - \mathbf{H})] \\ &= \sigma^2(\mathbf{I} - \mathbf{H}) \end{aligned} \tag{28.5}$$

and $Var(e_i) = \sigma^2(1 - h_{ii})$, which will be large when $h_{ii} \simeq 1/n$, and close to zero if $h_{ii} \simeq 1$. As the mean of the residuals is zero if the variance of e_i is very small this implies that its value will be close to zero, whatever the value of u_i.

The problem that each residual has a different variance leads to the definition of the standardized residuals, given by

$$r_i = \frac{e_i}{\widehat{s}_{\mathrm{R}}\sqrt{1 - h_{ii}}} \tag{28.6}$$

which will have variance equal to one. A third type of useful residuals are the predictive, deleted, or out-of-sample residuals, defined by $e_{(i)} = y_i - \widehat{y}_{i(i)}$, where $\widehat{y}_{i(i)}$ is computed in a sample with the i-th observation deleted. It can be shown that

$$e_{(i)} = \frac{e_i}{(1 - h_{ii})} \tag{28.7}$$

and the variance of these predictive residuals is $\sigma^2/(1 - h_{ii})$. If we estimate σ^2 by $\widehat{s}^2_{\mathrm{R}(i)}$, the residual variance in a regression which does not include the i-th observation, the standardization of the predictive residual leads to the Studentized residual, defined by

$$\widehat{t}_i = \frac{e_i}{\widehat{s}_{\mathrm{R}(i)}\sqrt{1 - h_{ii}}} \tag{28.8}$$

which has a Student t distribution with $n - p - 1$ degrees of freedom. An alternative useful expression of these residuals is based on $h_{ii(i)} = \boldsymbol{x}_i'(\mathbf{X}_{(i)}\mathbf{X}_{(i)})^{-1}\boldsymbol{x}_i = h_{ii}/(1 - h_{ii})$, where $\mathbf{X}_{(i)}$ is the $(n-1) \times p$ matrix without the row $\boldsymbol{x}_i'$, and therefore, we have the alternative expression:

$$\widehat{t}_i = \frac{e_{(i)}}{\widehat{s}_{\mathrm{R}(i)}\sqrt{1 + h_{ii(i)}}}. \tag{28.9}$$

28.2 Diagnosis for a Single Outlier

28.2.1 Outliers

If one observation, y_h, does not follow the regression model, either because its expected value is not $\boldsymbol{x}_h'\boldsymbol{\beta}$, or its conditional variance is not σ^2, we will say that it is an outlier. These discrepancies are usually translated to the residuals. For instance, if the observation has been generated by a different model, $g(\boldsymbol{x}_h') + u_h$, then

$$e_h = g(\boldsymbol{x}_h') - \boldsymbol{x}_h'\hat{\boldsymbol{\beta}} + u_h$$

and the deviation $\left|g(\boldsymbol{x}_h')-\boldsymbol{x}_h'\hat{\boldsymbol{\beta}}\right|$ will be larger than $\left|\boldsymbol{x}_h'(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\right|$. However, we may not detect this observation because of the key role of the variable $\boldsymbol{x}_h'$. Suppose, in order to simplify, that we write $g(\boldsymbol{x}_h')=\boldsymbol{x}_h'\boldsymbol{\alpha}$, that is, the data is also generated by a linear model but with different parameter values. Then, even if $\boldsymbol{\alpha}$ is very different from $\boldsymbol{\beta}$, the size of $\left|\boldsymbol{x}_h'(\boldsymbol{\alpha}-\hat{\boldsymbol{\beta}})\right|$ depends on $\boldsymbol{x}_h'$ and the discrepancy between the parameter values would be easier to detect when $\left|\boldsymbol{x}_h'\right|$ is large than when it is small.

When the observation is an outlier because it has a measurement error which comes from a distribution with variance $k\sigma^2$, where $k>1$, we expect that $|u_h|$ will be larger than the rest of the measurement errors. It is intuitive, and it has been formally shown [28.34], that we cannot differentiate between a change in the mean and a change in the variance by using just one observation; also models which assume a change in the variance are equivalent to those which assume shifts in the mean of the observations. Thus, we consider a simple mean-shift model for a single outlier

$$y_h=\boldsymbol{x}_h'\boldsymbol{\beta}+w+u_h,$$

where w is the size of the outlier and u_h is $N(0,\sigma^2)$. A test for outliers can be made by estimating the parameter w in the model

$$y_i=\boldsymbol{x}_i'\boldsymbol{\alpha}+wI_i^{(h)}+u_i,\quad i=1,..,n,$$

where $I_i^{(h)}$ is a dummy variable given by $I_i^{(h)}=1$, when $i=h$ and $I_i^{(h)}=0$, otherwise. We can test for outliers by fitting this model for $h=1,\ldots,n$, and checking if the estimated coefficient $\widehat{w}$ is significant. It is easy to show that:

1. $\widehat{\boldsymbol{\alpha}}=(\mathbf{X}_{(i)}'\mathbf{X}_{(i)})^{-1}\mathbf{X}_{(i)}'\boldsymbol{y}_{(i)}=\widehat{\boldsymbol{\beta}}_{(i)}$, the regression parameters are estimated in the usual way, but deleting case $(y_j,\boldsymbol{x}_j)$;
2. $\widehat{w}=y_h-\boldsymbol{x}_h'\widehat{\boldsymbol{\alpha}}$, and therefore the estimated residual at this point, $e_h=y_h-\boldsymbol{x}_h'\widehat{\boldsymbol{\alpha}}-\widehat{w}=0$.
3. The t statistic to check if the parameter $\widehat{w}$ is significant is equal to the Studentized residual, t_h, as defined in (28.8).

Assuming that only one observation is an outlier the test is made by comparing the standardized residual to the maximum of a t distribution with $n-p-2$ degrees of freedom. Often, for moderate n, cases are considered as outliers if their Studentized residuals are larger than 3.5.

28.2.2 Influential Observations

An intuitive way to measure the effect of an observation on the estimated parameters, or on the forecasts, is to delete this observation from the sample and see how this deletion affects the vector of parameters or the vector of forecasts. A measure of the influence of the $i-$th observation on the parameter estimate is given by:

$$D(i)=\frac{(\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_{(i)}')\mathbf{X}'\mathbf{X}(\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_{(i)})}{p\widehat{s}_{\mathrm{R}}^2},\tag{28.10}$$

which, as the covariance of $\widehat{\boldsymbol{\beta}}$ is $\widehat{s}_{\mathrm{R}}^2(\mathbf{X}'\mathbf{X})^{-1}$, measures the change between $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}_{(i)}$ with relation to the covariance matrix of $\widehat{\boldsymbol{\beta}}$, standardized by the dimension of the vector p. This measure was introduced by *Cook* [28.16]. Of course other standardizations are possible. *Belsley* et al. [28.9] propose using $\widehat{s}_{\mathrm{R}(i)}^2$, the variance of the regression model when the ith observation is deleted, instead of $\widehat{s}_{\mathrm{R}}^2$, and *Diaz-García* and *Gonzalez-Farias* [28.35] have suggested standardizing the vector $(\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_{(i)})$ by its variance, instead of using the variance of $\widehat{\boldsymbol{\beta}}$. See *Cook*, *Peña* and *Weisberg* [28.36] for a comparison of some of these possible standarizations.

Equation (28.10) can also be written as the standardized change in the vector of forecasts:

$$D_i=\frac{\left(\widehat{\boldsymbol{y}}-\widehat{\boldsymbol{y}}_{(i)}\right)'\left(\widehat{\boldsymbol{y}}-\widehat{\boldsymbol{y}}_{(i)}\right)}{p\widehat{s}_{\mathrm{R}}^2},\tag{28.11}$$

where $\widehat{\boldsymbol{y}}_{(i)}=\mathbf{X}\widehat{\beta}_{(i)}=(\hat{y}_{1(i)},\ldots,\hat{y}_{n(i)})'$. Note that from (28.2) we have that $Var(\widehat{y}_i)=\sigma^2h_{ii}$ and as the average value of h_{ii} is p/n, (28.11) is standardized by this average value and by the dimension n of the vector. A third way to measure the influence of the ith point is to compare $\widehat{y}_i$ with $\widehat{y}_{(i)}$, where $\widehat{y}_{(i)}=\boldsymbol{x}_i'\widehat{\boldsymbol{\beta}}_{(i)}$. With the usual standardization by the variance we have:

$$D_i=\frac{\left(\widehat{y}_i-\widehat{y}_{(i)}\right)^2}{p\widehat{s}_R^2h_{ii}}\tag{28.12}$$

and, using the relation between the inverse of $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}_{(i)}'\mathbf{X}_{(i)}$, we obtain

$$\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{(i)}=(\mathbf{X}'\mathbf{X})^{-1}\boldsymbol{x}_i\frac{e_i}{1-h_{ii}}.\tag{28.13}$$

Inserting this into (28.10) it is easy to see that (28.12) is equivalent to (28.10) and (28.11). Also, as from (28.13) we have that

$$\widehat{\boldsymbol{y}}-\widehat{\boldsymbol{y}}_{(i)}=\boldsymbol{h}_i\frac{e_i}{1-h_{ii}},\tag{28.14}$$

where $\boldsymbol{h}_i$ is the $i-$th column of the $\mathbf{H}$ matrix, inserting this expression into (28.11) we obtain a convenient expression for the computation of Cook's statistics:

$$D_i = \frac{r_i^2 h_{ii}}{p(1-h_{ii})}, \tag{28.15}$$

where r_i is the standardized residual given by (28.6). For large n, the expected value of D_i can be approximated by

$$E(D_i) \simeq \frac{h_{ii}}{p(1-h_{ii})}, \tag{28.16}$$

and it will be very different for observations with different leverage.

Cook proposed judging the values of D_i by an $F(p; n-p; 1-\alpha)$, where F is the distribution used in building a confidence region for the $\boldsymbol{\beta}$ parameters. Thus, we may identify points as influential when they are able to move the estimate out of the confidence region for a fixed value of α and declare as influential those observations which verify $D_i \geq F(p; n-p; 1-\alpha)$. This solution is not satisfactory for large sample sizes because it is difficult for any observation to be deemed influential. *Muller* and *Mok* [28.37] have obtained the distribution of the D_i for normal explanatory variables, but this distribution is complicated.

Cook [28.17] proposed a procedure for the assessment of the influence on a vector of parameters θ of minor perturbation of a statistical model. This approach is very flexible and can be used to see the effect of small perturbations which would not normally be detected by deletion of one observation. He suggested introducing a $n \times p$ vector $\boldsymbol{w}$ of case weights and use the likelihood displacement $[L(\hat{\theta}) - L(\hat{\theta}_w)]$, where $\hat{\theta}$ is the maximum likelihood (ML) estimator of $\hat{\theta}$, and $\hat{\theta}_w$ is the ML when the case weight $\boldsymbol{w}$ is introduced. Then, he showed that the directions of greatest local change in the likelihood displacement for the linear regression model are given by the eigenvectors linked to the largest eigenvalues of the curvature matrix, $\mathbf{L} = \mathbf{EHE}$, where $\mathbf{E}$ is the vector of residuals. Later, we will see how this approach is related to some procedures for multiple-outlier detection.

28.2.3 The Relationship Between Outliers and Influential Observations

An outlier may or may not be an influential observation and an influential observation may or may not be an outlier. To illustrate this point consider the data in Table 28.1. We will use these data to build four data sets. The first includes cases 1–9 repeated three times, and has sample size $n = 27$. The other three are formed by adding a new observation to this data set. The set (a) is built by adding case 28(a), the set (b) by adding case 28(b) and the set (c) by adding case 28(c). Table 28.2 shows some statistics of these four data sets where (0) refers to the set of 27 observations and (a), (b) and (c) to the sets of 28 observations as defined before. The table gives the values of the estimated parameters, their t statistics in parentheses, the residual standard deviation, the leverage of the added point, the standardized residual for the added point and the value of Cook's statistics.

In set (a) observation 28 is clearly an outlier with a value of the standardized residual of 4.68, but it is not influential, as $D_{28}(a) = 0.92$, which is a small value. In case (b) the 28-th point is not an outlier, as $r_{28}(b) = 1.77$ is not significant, but it is very influential, as indicated by the large D_{28} value. Finally, in set (c) the observation is both an outlier, $r_{28} = 4.63$, and very influential, $D_{28} = 13.5$.

Table 28.1 Three sets of data which differ in one observation

Case	1	2	3	4	5	6	7	8	9	(a)	(b)	(c)
x_1	−2	0	2	−4	3	1	−3	−1	4	0	−3	−3
x_2	6.5	7.3	8.3	6.0	8.8	8.0	5.9	6.9	9.5	7.2	9	7.3
y	−1.5	0.5	1.6	−3.9	3.5	0.8	−2.7	−1.3	4.1	5	−1.5	4

Table 28.2 Some statistics for the three regressions fitted to the data in Table 28.1

	$\widehat{\beta}_0$	$t(\widehat{\beta}_0)$	$\widehat{\beta}_2$	$t(\widehat{\beta}_2)$	$\widehat{\beta}_1$	$t(\widehat{\beta}_1)$	$\widehat{s}_R$	h_{28}	r_{28}	D_{28}
(0)	2.38	(0.82)	−0.30	(0.78)	1.12	(6.24)	0.348	–	–	–
(*a*)	13.1	(1.7)	−1.72	(−1.66)	1.77	(3.69)	0.96	0.11	4.68	0.92
(*b*)	−2.74	(−2.9)	0.38	(3.08)	0.80	(13.87)	0.36	0.91	1.77	11.1
(*c*)	−25.4	(−5.41)	3.43	(5.49)	−0.624	(2.22)	0.91	0.65	4.63	13.5

Note that if the leverage is small $h_{ii} \simeq 1/n$, $h_{ii}/(1-h_{ii}) \simeq (n-1)^{-1}$, and by (28.15):

$$D_i = \frac{r_i^2}{p}\left(\frac{1}{n-1}\right),$$

then, if n is large, the observation cannot be influential, whatever the value of r_i^2. On the other hand, high-leverage observations with h_{ii} close to one will have a ratio $h_{ii}/(1-h_{ii})$ that is arbitrarily large and, even if r_i^2 is small, will be influential.

28.3 Diagnosis for Groups of Outliers

The procedures that we have presented in the previous section are designed for a single outlier. We can extend these ideas to multiple outliers as follows. Let I be an index set corresponding to a subset of r data points. The checking of this subset can be done by introducing dummy variables as in the univariate case. Assuming normality, the F test for the hypothesis that the coefficients of the dummy variables are zero is given by

$$F_{r,(n-p-r)} = \frac{\boldsymbol{e}_I'(\mathbf{I}-\mathbf{H}_I)^{-1}\boldsymbol{e}_I}{r\widehat{s}^2_{\mathrm{R}(I)}}$$

where $\boldsymbol{e}_I$ is the vector of least-squares residuals, $\mathbf{H}_I$ the $r\times r$ submatrix of $\mathbf{H}$, corresponding to the set of observations included in I, and $\widehat{s}^2_{\mathrm{R}(I)}$ the residual variance of the regression with the set I deleted. *Cook* and *Weisberg* [28.11] proposed to measure the joint influence of the data points with index in I by deleting the set I and computing, as in the single outlier case,

$$D_I = \frac{(\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_{(I)}')\mathbf{X}'\mathbf{X}(\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_{(I)})}{p\widehat{s}^2_{\mathrm{R}}},$$

which can also be written as a generalization of (28.15) by $D_I = [\boldsymbol{e}_I'(\mathbf{I}-\mathbf{H}_I)^{-1}\mathbf{H}_I(\mathbf{I}-\mathbf{H}_I)^{-1}\boldsymbol{e}_I]/p\widehat{s}^2_{\mathrm{R}}$. Note that a large value of D_I may be due to a single influential observation included in the set I or a sum of small individual effects of a set of observations that are masking each other. However, in the first case this single observation will be easily identified. Also, a subset of individually highly influential points, whose effect is to cancel each other out, will lead to a small value of D_I; again in this case, the individual effects will be easy to identify. However, to build this measure we should compute all sets of I in the n data, which would be impossible for large I and n.

The procedures for finding multiple outliers in regression can be divided into three main groups. The first is based on robust estimation. If we can compute an estimate that is not affected by the outliers, we can then find the outliers as those cases with large residuals with respect to the robust fit. We present briefly here the least median of squares (LMS) estimate proposed by *Rousseeuw* [28.38], which is used as an initial estimate in some diagnostic procedures based on a clean set, which we will review below. *Rousseeuw* [28.38] proposed generating many possible values of the parameters, $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N$, finding the residuals associated with each parameter value, $\boldsymbol{e}_i = \boldsymbol{y} - \mathbf{X}\,\boldsymbol{\beta}_i$ $(i = 1, .., N)$, and using the median of these residuals as a robust scale

$$s(\boldsymbol{\beta}_i) = median(e_{1i}^2, \ldots, e_{ni}^2). \qquad (28.17)$$

The value $\boldsymbol{\beta}_i$ that minimizes this robust scale is the LMS estimate. *Rousseeuw* [28.38] generates the parameter values $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N$ by resampling, that is, by taking many random samples of size p, $(\mathbf{X}_i, \boldsymbol{y}_i)$, where the matrix $\mathbf{X}_i$ is $p\times p$ and $\boldsymbol{y}_i$ is $p\times 1$, and computing the least-squares estimate (LSE) for each sample, $\boldsymbol{\beta}_i = \mathbf{X}_i^{-1}\boldsymbol{y}_i$. The LMS, although very robust, is not very efficient, and many other robust methods have been proposed to keep high robustness and achieve better efficiency in regression [28.31].

A second class of procedures uses robust ideas to build an initial clean subset and then combine least-squares estimates in clean subsets and diagnosis ideas for outlier detection. Three procedures in this spirit will be presented next; they can be very effective when p and n are not large.

For large data sets with many predictors and high-leverage observations, robust estimates can be very difficult to compute and procedures based on the clean-set idea may not work well, because of the difficulty in selecting the initial subset. The third type of procedures are based on eigenstructure analysis of some diagnostic matrices and are especially useful for large data sets.

28.3.1 Methods Based on an Initial Clean Set

Kianifard and *Swallow* [28.26, 27] proposed to build a *clean* set of observations and check the rest of the data with respect to this set. If the observation closest to the clean set is not an outlier, then the clean set is increased by one observation, and continue to do so until no new

observations can be incorporated into the basic set. The key step in this procedure is to find the initial subset, because if it contains outliers the whole procedure breaks down. These authors proposed using either the predictive or standardized residuals, or a measure of influence such as D_i.

A similar procedure was proposed by *Hadi* and *Simonoff* [28.1, 2]. In [28.2] they recommend building the initial subset using the LMS. The clean set is built by computing this robust estimate and then uses the $h = \left(\frac{n+p+1}{2}\right)$ observations with the smallest residuals with respect to this robust fit to form the initial clean set, which we call M. The procedure continues by fitting a regression model by least squares to this clean set M. Calling $\widehat{\boldsymbol{\beta}}_M$ the estimated LSE parameters and $\widehat{\sigma}_M$ the residual standard deviation, a set of in-sample and out-of-sample residuals is obtained as follows

$$d_i = \frac{\left|y_i - \boldsymbol{x}_i'\boldsymbol{\beta}_M\right|}{\widehat{\sigma}_M\sqrt{1-\boldsymbol{x}_i'(\mathbf{X}_M'\mathbf{X}_M)^{-1}\boldsymbol{x}_i}}, \quad \text{if } i \in M,$$

$$d_i = \frac{\left|y_i - \boldsymbol{x}_i'\boldsymbol{\beta}_M\right|}{\widehat{\sigma}_M\sqrt{1+\boldsymbol{x}_i'(\mathbf{X}_M'\mathbf{X}_M)^{-1}\boldsymbol{x}_i}}, \quad \text{if } i \notin M.$$

That is, d_i represents the standardized residual (28.6) for the data in set M and the predictive residual (28.9) for observations outside this set. Then, all of the observations are arranged in increasing order according to d_i. Let s be the size of the set M (which is h in the first iteration, but will change as explained below). If $d_{(s+1)}$ is smaller than some critical value, a new set of size $s+1$ is built with the $s+1$ observations with smallest d values. If $d_{(s+1)}$ is larger than some critical value, all observations out of the set M are declared as outliers and the procedure stops. If $n = s+1$ we stop and declare that there are no outliers in the data. These authors proposed using as critical values those of the t distribution adjusted by Bonferroni, that is $t\left(\frac{\alpha}{2(s+1)}, s-p\right)$.

Atkinson [28.3] proposed a similar approach called the forward search. His idea is again to combine a robust estimate with diagnostic analysis. He computes the LMS estimate but, instead of generating a large set of candidates by random sample, he generates a set of candidate values for $\widehat{\boldsymbol{\beta}}$ by fitting least-squares subsamples of size p, $p+1, \ldots, n$. The procedure is as follows. We start by generating a random sample of size p; let I_p be the indices of the observations selected. Then, we compute the parameters $\widehat{\boldsymbol{\beta}}(p)$ by LSE, and the residual for all cases, $\boldsymbol{e} = \boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}(p)$. The residuals are corrected by

$$u_i^2 = e_i^2, \qquad i \in I \tag{28.18}$$

$$u_i^2 = e_i^2/(1+h_{ii}), \qquad i \notin I$$

and these residuals u_i^2 are ordered and the smallest $p+1$ are selected. With this new sample of size $m = p+1$ the process is repeated, that is, the parameters are computed by LSE and the residuals to this fit for the n points are obtained. The corrected residuals (28.18) are computed and the process is continued. In this way we obtain a set of estimates, $\widehat{\boldsymbol{\beta}}(m), m = p, .., n$, the corresponding residuals, $\boldsymbol{e}(m) = \boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}(m)$, and the robust scales (28.17), $s[\widehat{\boldsymbol{\beta}}(m)]$. The value selected is the $\widehat{\boldsymbol{\beta}}(m)$ which minimizes the robust scale. This process is a complete forward search and several forward searches are done starting with different random samples. The residuals are then identified by using this LMS estimate computed from several forward searches. An improvement of this procedure was proposed by *Atkinson* and *Riani* [28.15], which clearly separates the estimation of the clean subset and the forward search. The initial estimate is computed, as proposed by *Rousseeuw* [28.38], by taking many random samples of size p. The forward search is then applied, but stressing the use of diagnostic statistics to monitor the performance of the procedure.

Finally, *Swallow* and *Kianifard* [28.4] also suggested a similar procedure, which uses a robust estimate of the scale and determines the cutoff values for testing from simulations.

These procedures work when both p and n are not large and the proportion of outliers is moderate, as shown in the simulated comparison by *Wisnowski* et al. [28.39]. However, they do not work as well in large data sets with high contamination. The LMS estimates rely on having at least a sample of size p without outliers, and we need an unfeasible number of samples to have a large probability of this event when p and n are large [28.6]. This good initial estimate is the key for procedures based on clean sets. In the next section we will present procedures that can be applied to large data sets.

28.3.2 Analysis of the Influence Matrix

Let us define the matrix of forecast changes, as the matrix of changes in the forecast of one observation when another observation is deleted. This matrix is given by

$$\mathbf{T} = \begin{pmatrix} \widehat{y}_1 - \widehat{y}_{1(1)} & \widehat{y}_1 - \widehat{y}_{1(2)} & \ldots & \widehat{y}_1 - \widehat{y}_{1(n-1)} & \widehat{y}_1 - \widehat{y}_{1(n)} \\ \widehat{y}_2 - \widehat{y}_{2(1)} & \widehat{y}_2 - \widehat{y}_{2(2)} & \ldots & \widehat{y}_2 - \widehat{y}_{2(n-1)} & \widehat{y}_2 - \widehat{y}_{2(n)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \widehat{y}_{n-1} - \widehat{y}_{n-1(1)} & \widehat{y}_n - \widehat{y}_{n(2)} & \ldots & \widehat{y}_n - \widehat{y}_{n(n-1)} & \widehat{y}_n - \widehat{y}_{n(n)} \\ \widehat{y}_n - \widehat{y}_{n(1)} & \widehat{y}_n - \widehat{y}_{n(2)} & \ldots & \widehat{y}_n - \widehat{y}_{n(n-1)} & \widehat{y}_n - \widehat{y}_{n(n)} \end{pmatrix}.$$

The columns of this matrix are the vectors $\boldsymbol{t}_i = \hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(i)}$, and Cook's statistic is their standardized norm. These

vectors can also be written as $\boldsymbol{t}_i = \boldsymbol{e}_{(i)} - \boldsymbol{e}$, where $\boldsymbol{e}_{(i)}$ is the vector of residuals when the i-th observation is deleted. Therefore, $\mathbf{T}$ can also be considered the matrix of residual changes. *Peña* and *Yohai* [28.5] define the $n \times n$ influence matrix $\mathbf{M}$ as

$$\mathbf{M} = \frac{1}{ps_{\mathrm{R}}^2}\mathbf{T}'\mathbf{T}.$$

As $\mathbf{H}$ is idempotent it can be shown immediately that $\mathbf{M}$ is given by

$$\mathbf{M} = \frac{1}{ps_{\mathrm{R}}^2}\mathbf{EDHDE}, \tag{28.19}$$

where $\mathbf{E}$ is a diagonal matrix with the residuals on the main diagonal, and $\mathbf{D}$ is a diagonal matrix with elements $(1-h_{ii})^{-1}$. By (28.7) $\mathbf{ED}$ is the diagonal matrix of predictive residuals. Therefore, the ij-th element of $\mathbf{M}$, is

$$m_{ij} = \frac{e_i e_j h_{ij}}{(1-h_{ii})(1-h_{jj})ps_{\mathrm{R}}^2} = \frac{e_{i(i)}e_{j(j)}h_{ij}}{ps_{\mathrm{R}}^2}\,.$$

Assuming that all the residuals are different from zero, from (28.4) the rank of $\mathbf{M}$ is equal to p, the rank of $\mathbf{H}$. Observe that the diagonal elements of $\mathbf{M}$ are the Cook's statistics.

Let $r_{ij} = m_{ij}/m_{ii}^{1/2}m_{jj}^{1/2}$ be the uncentered correlation coefficient between $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$. Let us show that the eigenvectors of the matrix $\mathbf{M}$ will be able to indicate groups of influential observations. Suppose that there are k groups of influential observations $I_1, \ldots, I_k$, such that

1. If $i, j \in I_h$, then $|r_{ij}| = 1$. This means that the effects on the least-squares fit produced by the deletion of two points in the same set I_h have correlation 1 or -1.
2. If $i \in I_j$ and $l \in I_h$ with $j \neq h$, then $r_{il} = 0$. This means that the effects produced on the least-squares fit by observations i and j belonging to different sets are uncorrelated.
3. If i does not belong to any I_h, then $m_{ij} = 0$ for all j. This means that data points outside these groups have no influence on the fit.

Now, according to (1) we can split each set I_h into I_h^1 and I_h^2 such that: (1) if $i, j \in I_h^q$, then $r_{ij} = 1$; (2) if $i \in I_h^1$ and $j \in I_h^2$, then $r_{ij} = -1$. Let $\boldsymbol{v}_1 = (v_{11}, \ldots, v_{1n})', \ldots, \boldsymbol{v}_k = (v_{k1}, \ldots, v_{kn})'$ be defined by $v_{hj} = m_{jj}^{1/2}$ if $j \in I_h^1$; $v_{hj} = m_{jj}^{1/2}$ if $j \in I_h^1$; $v_{hj} = -m_{jj}^{1/2}$ if $j \in I_h^2$ and $v_{hj} = 0$ if $j \notin I_h$. Then, if (1)–(3) hold, by (28.6) the matrix $\mathbf{M}$ is

$$\mathbf{M} = \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i',$$

and since the $\boldsymbol{v}_i$ are orthogonal, the eigenvectors of $\mathbf{M}$ are $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$, and the corresponding eigenvalues $\lambda_1, \ldots, \lambda_k$ are given by

$$\lambda_h = \sum_{i \in I_h} m_{ii}.$$

It is clear that, when the matrix $\mathbf{M}$ satisfies (1)–(3), the only sets I with large C_I are I_h^q, $1 \leq h \leq k$, $q = 1, 2$, and these sets may be found by looking at the eigenvectors associated with non-null eigenvalues of $\mathbf{M}$. Note that (28.6) can also be written as

$$r_{ij} = \mathrm{sign}(e_i)\mathrm{sign}(e_j)h_{ij}/(h_{ii}h_{jj})^{1/2},$$

which means that, in the extreme case that we have presented, the $\mathbf{H}$ matrix and the signs of the residuals are able, by themselves, to identify the set of points that are associated with masking. For real data sets, (1)–(3) do not hold exactly. However, the masking effect is typically due to the presence of blocks of influential observations in the sample having similar or opposite effects. These blocks are likely to produce a matrix $\mathbf{M}$ with a structure close to that described by (1)–(3). In fact, two influential observations i, j producing similar effects should have r_{ij} close to 1, and close to -1 when they have opposed effects. Influential observations with non-correlated effects have $|r_{ij}|$ close to 0. The same will happen with non-influential observations. Therefore, the eigenvectors will have approximately the structure described above, and the null components will be replaced by small values.

This suggests that we should find the eigenvectors corresponding to the p non-null eigenvalues of the influence matrix $\mathbf{M}$, consider the eigenvectors corresponding to large eigenvalues, and define the sets I_j^1 and I_j^2 by those components with large positive and negative weights, respectively. *Peña* and *Yohai* [28.5] proposed the following procedure.

Step 1: *Identifying sets of outlier candidates*. A set of candidate outlier is obtained by analyzing the eigenvectors corresponding to the non-null eigenvalues of the influence matrix $\mathbf{M}$, and by searching in each eigenvector for a set of coordinates with relatively large weight and the same sign.

Step 2: *Checking for outliers*. (a) Remove all candidate outliers. (b) Use the standard F and t statistics to

Table 28.3 A simulated set of data

	1	2	3	4	5	6	7	8	9(a)	10(a)	9(b)	10(b)	9(c)	10(c)
x	1	2	3	4	5	6	7	8	12	12	12	12	12	12
y	2.0	2.9	3.9	5.1	6.2	6.9	7.8	9.1	19	20	19	7	13	7

Table 28.4 Eigen-analysis of the influence matrix for the data from Table 28.3. The eigenvectors and eigenvalues are shown

	λ_1	λ_1/λ_2	1	2	3	4	5	6	7	8	9	10
(a)	1.27	2.87	−0.17	−0.06	−0.00	−0.00	−0.02	−0.10	−0.22	−0.33	0.42	0.79
(b)	3.78	3.783	0.00	−0.00	−0.00	−0.00	−0.00	0.00	−0.00	−0.00	−0.71	0.71
(c)	3.25	32	−0.05	−0.02	−0.00	−0.00	−0.01	−0.02	−0.04	−0.10	−0.50	0.85

test for groups or individual outliers. Reject sets or individual points with F or t statistics larger than some constant c. For the F statistic the c value corresponds to the distribution of the maximum F over all sets of the same size, and this distribution is unknown. Therefore, it is better to use the t statistic and choose the c value by the Bonferroni inequality or, better still, by simulating the procedure with normal errors. (c) If the number of candidate outliers is larger than $n/2$, the previous procedure can be applied separately to the points identified in each eigenvector.

As an illustration we will use the simulated data from Table 28.3, which are plotted in Fig. 28.1.

The three sets of data have in common cases 1–8 and differ in cases 9 and 10. In the first set of data the largest values of the Cook's statistics are $D_{10} = 0.795$, $D_1 = 0, 29$ and $D_9 = 0.228$. The most influential observation is the 10-th, which has a standardized residual $r_{10} = 1.88$, thus there is no evidence that the point is an outlier. However, the first eigenvector of the influence matrix leads to the results shown in Table 28.4. We see that both cases 9 and 10 appear separated from the rest. When they are deleted from the sample and checked against the first eight observations we obtain the values indicated in Table 28.5, where they are clearly declared as outliers. Thus, in this example the eigenvalues of the influence matrix are able to avoid the masking effect which was clearly present in the univariate statistics.

In case (b), as both outliers have a different sign, they do not produce masking, and both of them are detected by the univariate analysis: $D_9 = 1.889$, and $D_{10} = 1.893$, and the outlier tests are $t_{10} = 5.20$ and $t_9 = -5.24$. The two points are also shown in the extremes of the eigenvalue. Finally in case (c) there is only one outlier which is detected by both the univariate and multivariate analysis.

The influence matrix $\mathbf{M}$ may be considered a generalization of Cook's local influence matrix $\mathbf{L} = \mathbf{EHE}$ [28.17]. It replaces the matrix of residuals $\mathbf{E}$ by the matrix of standardized residuals $\mathbf{ED}$. If there are no high-leverage observations and the h_{ii} are similar for all points, both matrices will also be similar, and will have similar eigenvectors. However, when the observations have very different leverages, the directions corresponding to the eigenvectors of the matrix $\mathbf{M}$ give more weight to the influence of the high-leverage observations, which are

Table 28.5 Values of the t statistics for testing each point as an outlier

Case	9	10
(a)	27.69	32.28
(b)	31.94	−32.09
(c)	−0.07	−32.09

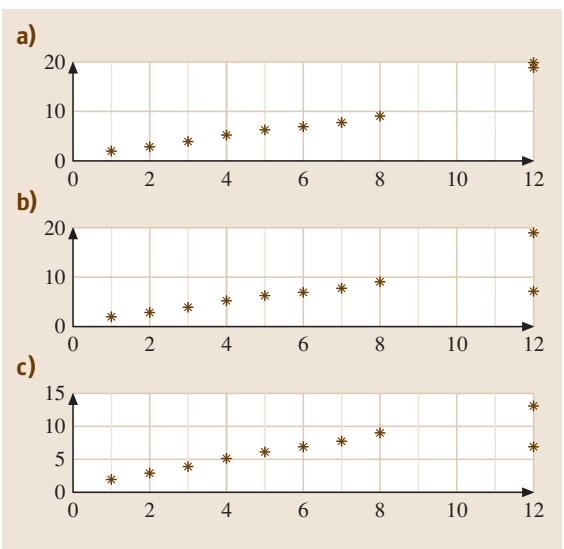


Fig. 28.1 The simulated data from Table 28.3

Table 28.6 Eigenvalues of the sensitivity matrix for the data from Table 28.3

	1	2	3	4	5	6	7	8	9	10
v_1	0.502	0.455	0.407	0.360	0.312	0.264	0.217	0.170	−0.020	−0.020
v_2	−0.191	−0.119	−0.046	0.026	0.099	0.172	0.245	0.318	0.610	0.610

precisely those that are more likely to produce masking effects.

Note that the rank of the influence matrix **M** is p, the same as the rank of **H**, and therefore we do not need to compute n eigenvectors as we only have p eigenvalues linked to nonzero eigenvalues. Thus, the procedure can be applied for very large data sets, see *Peña* and *Yohai* [28.5] for the details of the implementation.

28.3.3 The Sensitivity Matrix

If instead of looking at the columns of the matrix of forecast changes **T** we look at its rows, a different perspective appears. The rows indicate the sensitivity of each point, that is, how the forecast for a given point changes when we use as the sample the n sets of $n-1$ data built by deleting each point of the sample. In this way we analyze the sensitivity of a given point under a set of small perturbations of the sample. Let

$$\boldsymbol{s}_i = (\widehat{y}_i - \widehat{y}_{i(1)}, \ldots, \widehat{y}_i - \widehat{y}_{i(n)})'$$

be the i-th row of the matrix **T**. From (28.14) we can write

$$\boldsymbol{s}_i = (h_{i1}e_1/(1-h_{11}), \ldots, h_{in}e_n/(1-h_{nn})) = \mathbf{ED}\boldsymbol{h}_i,$$

where **E** and **D** are diagonal matrices of residuals and inverse leverage, respectively, defined in the previous section, and $\boldsymbol{h}_i$ is the i-th column of **H**. We define the sensitivity matrix by

$$\mathbf{P} = \frac{1}{p\widehat{s}_{\mathrm{R}}^2}\begin{pmatrix} \boldsymbol{s}_1'\boldsymbol{s}_1 & \ldots & \boldsymbol{s}_1'\boldsymbol{s}_n \\ \ldots & \ldots & \ldots \\ \boldsymbol{s}_1'\boldsymbol{s}_n & \ldots & \boldsymbol{s}_n'\boldsymbol{s}_n \end{pmatrix},$$

which can be computed by

$$\mathbf{P} = \frac{1}{p\widehat{s}_{\mathrm{R}}^2}\mathbf{HED}^2\mathbf{EH}, \tag{28.20}$$

and has elements

$$p_{ij} = \frac{1}{p\widehat{s}_{\mathrm{R}}^2}\sum_{k=1}^{n}\frac{e_k^2}{(1-h_{kk})^2}h_{ik}h_{jk}.$$

It can be shown that the sensitivity and the influence matrix have the same eigenvalues and we can obtain the eigenvectors of one matrix from the eigenvectors of the other. *Peña* and *Yohai* [28.6] have shown that eigenvectors of the sensitivity matrix are more powerful for the identification of groups of outliers than those of the influence matrix, although they often lead to the same results. These authors also show that these methods work very well for large sets with many predictors and high levels of contamination.

In the following example we show the use of this matrix for detecting groups of outliers. If we compute the eigenvectors of the sensitivity matrix for the data in Table 28.3 we obtain the results presented in Table 28.6. The first eigenvector clearly separates the observations 9 and 10 from the rest. In fact, if we order the coordinates of this vector we find the largest ratio at $170/20 = 8.5$ which separates cases 9 and 10 from the others.

28.4 A Statistic for Sensitivity for Large Data Sets

The analysis of the eigenvalues of the sensitivity matrix is a very powerful method for finding outliers. However, for large data sets it would be very convenient to have a simple statistic, fast to compute, which can be incorporated into the standard output of regression fitting and which could indicate groups of high-leverage outliers, which are the most difficult to identify. This statistic can be obtained through a proper standardization of the diagonal elements of the sensitivity matrix. *Peña* [28.7] defines the sensitivity statistic at the i-th observation S_i as the squared norm of the standardized vector $\boldsymbol{s}_i$, that is,

$$S_i = \frac{\boldsymbol{s}_i'\boldsymbol{s}_i}{p\widehat{Var}(\hat{y}_i)}, \tag{28.21}$$

and using (28.14) and $\widehat{Var}(\hat{y}_i) = \hat{s}^2_{\mathrm{R}} h_{ii}$, this statistic can be written as

$$S_i = \frac{1}{p\hat{s}^2_{\mathrm{R}} h_{ii}} \sum_{j=1}^{n} \frac{h^2_{ji} e^2_j}{(1-h_{jj})^2}. \tag{28.22}$$

An alternative way to write S_i, is as a linear combination of the sample Cook's distance. From (28.12) and (28.22), we have

$$S_i = \sum_{j=1}^{n} \rho^2_{ji} D_j, \tag{28.23}$$

where $\rho_{ij} = (h^2_{ij}/h_{ii}h_{jj})^{1/2} \leq 1$ is the correlation between forecasts $\hat{y}_i$ and $\hat{y}_j$. Also, using the predictive residuals, $e_{j(j)} = e_j/(1-h_{jj})$, we have that

$$S_i = \frac{1}{p\hat{s}^2_{\mathrm{R}}} \sum_{j=1}^{n} w_{ji} e^2_{j(j)} \tag{28.24}$$

and S_i is a weighted combination of the predictive residuals.

The sensitivity statistics has three interesting properties. The first is that, in a sample without outliers or high-leverage observations, all the cases have the same expected sensitivity, approximately equal to $1/p$. This is an important advantage over Cook's statistic, which has an expected value that depends heavily on the leverage of the case. The second property is that, for large sample sizes with many predictors, the distribution of the S_i statistic will be approximately normal. This again is an important difference from Cook's distance, which has a complicated asymptotic distribution [28.37]. This normal distribution allows the computation of cutoff values for finding outliers. The third property is that, when the sample is contaminated by a group of similar outliers with high leverage, the sensitivity statistic will discriminate between the outliers and the good points, and the sensitivity statistic S_i is expected to be smaller for the outliers than for the good data points.

These properties are proved in *Peña* [28.7]. The normality of the distribution of the S_i statistic implies that we can search for outliers by finding observations with large values of $[S_i - E(S_i)]/std(S_i)$. As the possible presence of outliers and high leverage points will affect the distribution of S_i, it is better to use robust estimates such as as the median or the median of the absolute deviations (MAD) from the sample median, and consider as heterogeneous observations those which satisfy:

$$|S_i - med(S)| \geq 4.5\mathrm{MAD}(S_i) \tag{28.25}$$

where $med(S)$ is the median of the S_i values and $\mathrm{MAD}(S_i) = med\,|S_i - med(S)|$. For normal data $\mathrm{MAD}(S_i)/.645$ is a robust estimate for the standard deviation and the previous rule is roughly equivalent to taking three standard deviations in the normal case. In *Peña* [28.7] it is shown that this statistic can be very useful for the diagnostic analysis of large data sets.

28.5 An Example: The Boston Housing Data

As an example of the usefulness of the sensitivity statistics and to compare it with the procedures based on eigenvalues, we will use the Boston housing data set which consists of 506 observations on 14 variables, available at Carnegie Mellon University, Department of Statistics, Pittsburgh (http://lib.stat.cmu.edu). This data set was given by *Belsley* et al. [28.9] and we have used the same variables they considered: the dependent variable is the logarithm of the median value of owner-occupied homes.

Figure 28.2 shows the diagnostic analysis of this data set. The first row corresponds to the residuals of the regression model. The residuals have been divided by their standard error and the first plot shows a few points which can be considered as outliers. The plot of the Studentized residual is similar and identifies the same points as outliers. The second row gives information about Cook's D statistics. There are clearly some points in the middle of the sample which are more influential than the rest, but all the values of the statistic are small and, as we expect a skewed distribution, the conclusion is not clear. However, the sensitivity statistics clearly identifies a group of extreme observations which are not homogeneous with the rest. The median of the sensitivity statistic is 0.0762, which is very close to the expected value $1/p = 1/14 = 0.0714$. The MAD is 0.0195 and the plot indicates that 45 observations are heterogeneous with respect to the rest. These observations are most of the cases 366–425 and some other isolated points. From *Belsley* et al. [28.9] we obtain that cases 357–488 correspond to Boston, whereas the rest correspond to the suburbs. Also, the 45 points indicated by the statistic S_i as outliers all correspond to some central districts of Boston, including the downtown area, which suggests that the relation among the variables could be different in these dis-

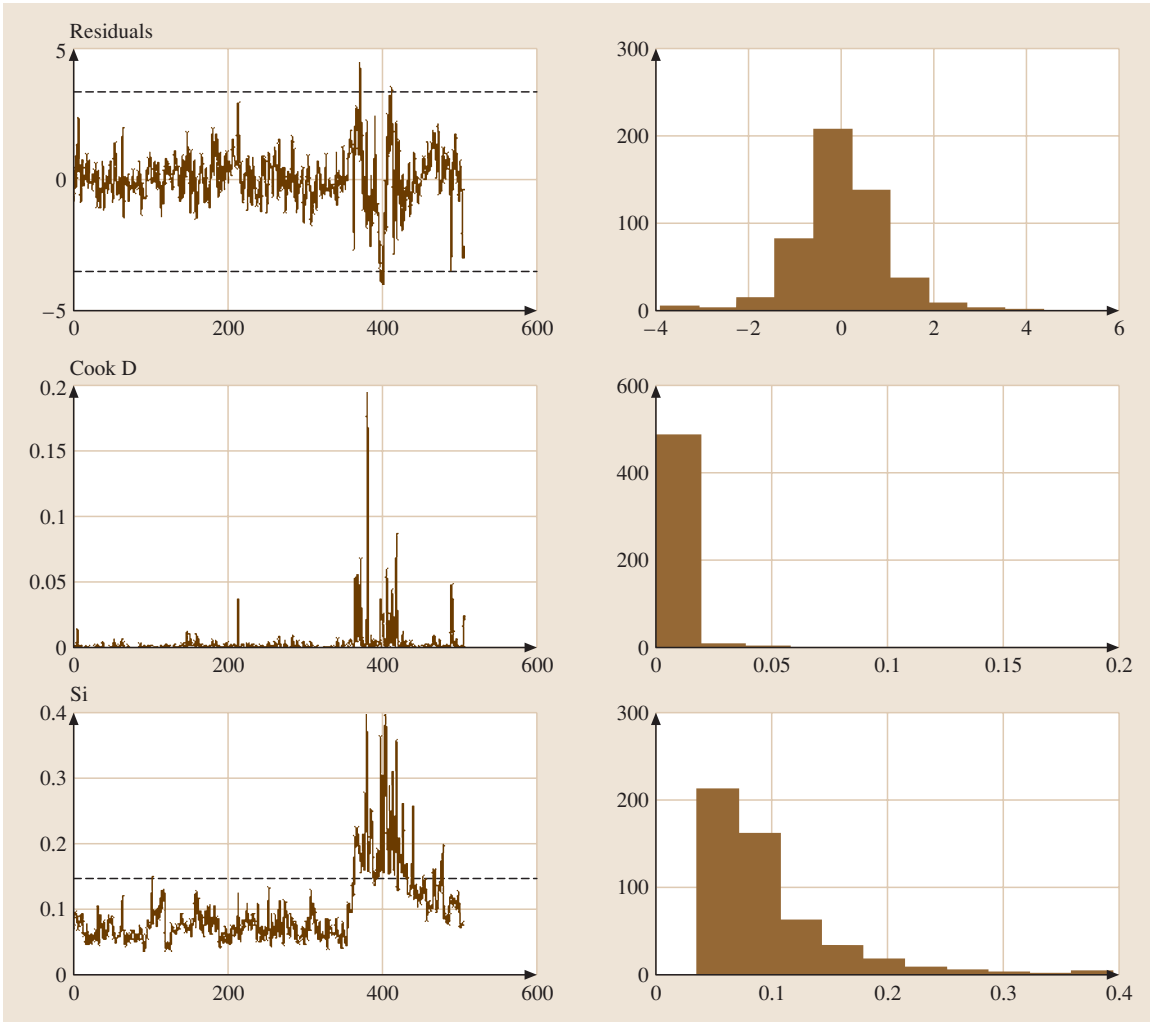


Fig. 28.2 Residuals, Cook's statistics and sensitivity statistics for the Boston housing data. *Right*, histogram; *left* case plot of the value of the statistic

tricts than in the rest of the sample. In fact, if we fit regression equations to these two groups we find very different coefficients for the regression coefficients in both groups of data, and in the second group only five variables are significant. Also, we obtain a large reduction in the residual sum of squares (RSE) when fitting different regression equations in the two groups.

Figure 28.3 shows the first eigenvalues of the matrix of influence and sensitivity. Although both eigenvectors indicate heterogeneity, the one from the matrix of sensitivity is more clear.

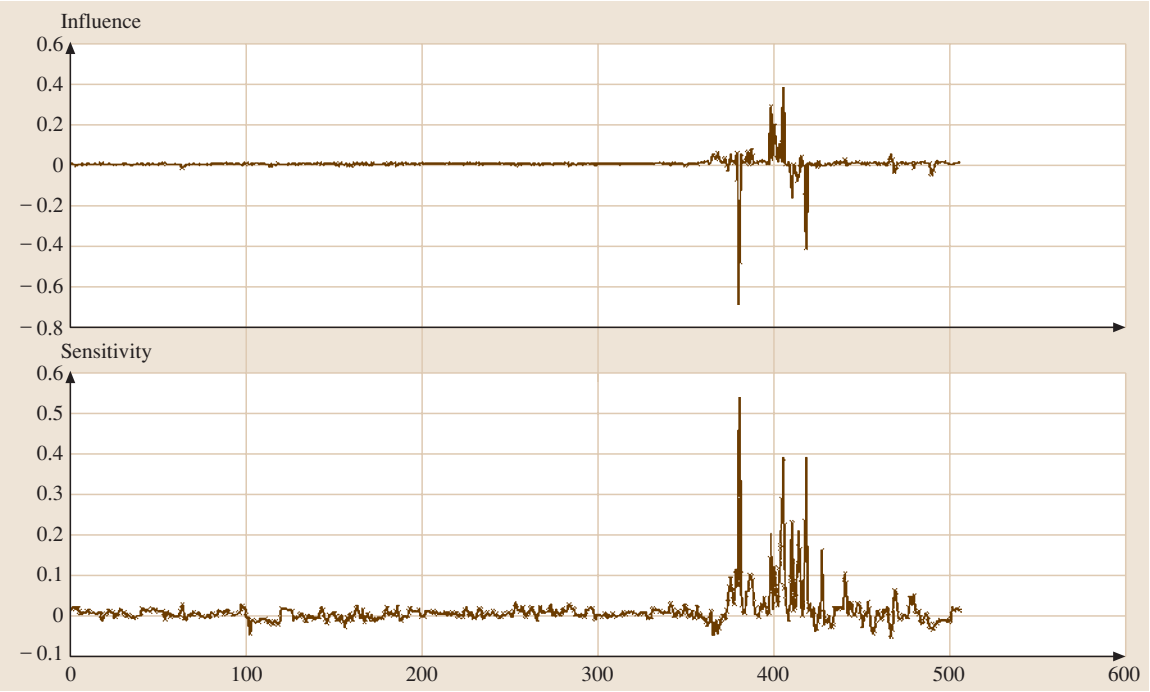


Fig. 28.3 First eigenvalue of the influence and sensitivity matrices

28.6 Final Remarks

We have shown different procedures for diagnosis in regression models and have stressed that the detection of groups of outliers in regression in large data sets can be made by eigen-analysis of the influence and sensitivity matrices. We have also shown that a single statistic of sensitivity is able to reveal masked outliers in many difficult situations. The most challenging problem today is to identify heterogeneity when we do not have a central model which explains more than 50% of the data and groups of outliers, as has been assumed in this article, but different regression models in different regions of the parameter space. In this case robust methods are no longer useful and we need other methods to solve this problem. A promising approach is the split and recombine (SAR) procedure, which has been applied to find heterogeneity in regression models by *Peña* et al. [28.40]. These situations are very close to cluster analysis and finding clusters around different regression lines is today a promising line of research.

References

28.1 A. S. Hadi, J. S. Simonoff: Procedures for the identification of multiple outliers in linear models, J. Am. Statist. Assoc. **88**, 1264–1272 (1993)

28.2 A. S. Hadi, J. S. Simonoff: Improving the estimation and outlier identification properties of the least median of squares and minimum volume ellipsoid estimators, Parisankhyan Samikkha **1**, 61–70 (1994)

28.3 A. C. Atkinson: Fast very robust methods for the detection of multiple outliers, J. Am. Statist. Assoc. **89**, 1329–1339 (1994)

28.4 W. Swallow, F. Kianifard: Using robust scale estimates in detecting multiple outliers in linear regression, Biometrics **52**, 545–556 (1996)

28.5 D. Peña, V. J. Yohai: The detection of influential subsets in linear regression using an influence matrix, J. R. Statist. Soc. B **57**, 145–156 (1995)

28.6 D. Peña, V.J. Yohai: A fast procedure for robust estimation and diagnostics in large regression problems, J. Am. Statist. Assoc. **94**, 434–445 (1999)

28.7 D. Peña: A new statistic for influence in linear regression, Technometrics **47**(1), 1–12 (2005)

28.8 G.E.P. Box: When Murphy speaks listen, Qual. Prog. **22**, 79–84 (1989)

28.9 D.A. Belsley, E. Kuh, R.E. Welsch: *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* (Wiley, New York 1980)

28.10 D.M. Hawkins: *Identification of Outliers* (Chapman Hall, New York 1980)

28.11 R.D. Cook, S. Weisberg: *Residuals and Influence in Regression* (Chapman Hall, New York 1982)

28.12 A.C. Atkinson: *Plots, Transformations and Regression* (Clarendon, Oxford 1985)

28.13 S. Chatterjee, A.S. Hadi: *Sensitivity Analysis in Linear Regression* (Wiley, New York 1988)

28.14 V. Barnett, T. Lewis: *Outliers in Statistical Data*, 3 edn. (Wiley, New York 1994)

28.15 A.C. Atkinson, M. Riani: *Robust Diagnostic Regression Analysis* (Springer, Berlin Heidelberg New York 2000)

28.16 R.D. Cook: Detection of influential observations in linear regression, Technometrics **19**, 15–18 (1977)

28.17 R.D. Cook: Assessment of local influence (with discussion), J. R. Statist. Soc. B **48**(2), 133–169 (1986)

28.18 G.C. Brown, A.J. Lawrence: Theory and ilustration of regression influence diagnostics, Commun. Statist. A **29**, 2079–2107 (2000)

28.19 M. Suárez Rancel, M.A. González Sierra: Regression diagnostic using local influence: A review, Commun. Statist. A **30**, 799–813 (2001)

28.20 G. Hartless, J. G. Booth, R.C. Littell: Local influence of predictors in multiple linear regression, Technometrics **45**, 326–332 (2003)

28.21 F. Critchley, R.A. Atkinson, G. Lu, E. Biazi: Influence analysis based on the case sensitivity function, J. R. Statist. Soc. B **63**(2), 307–323 (2001)

28.22 J. Lawrance: Deletion influence and masking in regression, J. R. Statist. Soc. B **57**, 181–189 (1995)

28.23 D.M. Hawkins, D. Bradu, G.V. Kass: Location of several oultiers in multiple regression data using elemental sets, Technometrics **26**, 197–208 (1984)

28.24 J.B. Gray, R.F. Ling: K–Clustering as a detection tool for influential subsets in regression, Technometrics **26**, 305–330 (1984)

28.25 M.G. Marasinghe: A multistage procedure for detecting several outliers in linear regression, Technometrics **27**, 395–399 (1985)

28.26 F. Kianifard, W. Swallow: Using recursive residuals calculated in adaptively ordered observations to identify outliers in linear regression, Biometrics **45**, 571–585 (1989)

28.27 F. Kianifard, W. Swallow: A Monte Carlo Comparison of five Procedures for Identifying Outliers in Lineal Regression, Commun. Statist. (Theory and Methods) **19**, 1913–1938 (1990)

28.28 A.C. Atkinson: Masking unmasked, Biometrika **73**, 533–41 (1986)

28.29 P. Huber: Between Robustness and Diagnosis. In: *Directions in Robust Statistics and Diagnosis*, ed. by W. Stahel, S. Weisberg (Springer, Berlin Heidelberg New York 1991) pp. 121–130

28.30 P.J. Rousseeuw, A.M. Leroy: *Robust Regression and Outlier Detection* (Wiley, New York 1987)

28.31 R.A. Maronna, R.D. Martin, V.J. Yohai: *Robust Statistics, Theory and Practice* (Wiley, New York 2006)

28.32 G.E.P. Box, C.G. Tiao: A Bayesian approach to some outlier problems, Biometrika **55**, 119–129 (1968)

28.33 A. Justel, D. Peña: Bayesian unmasking in linear models, Comput. Statist. Data Anal. **36**, 69–94 (2001)

28.34 D. Peña, I. Guttman: Comparing probabilistic models for outlier detection, Biometrika **80**(3), 603–610 (1993)

28.35 J.A. Diaz-Garcia, G. Gonzalez-Farias: A note on the Cook's distance, J. Statist. Planning Inference **120**, 119–136 (2004)

28.36 R.D. Cook, D. Peña, S. Weisberg: The likelihood displacement. A unifying principle for influence, Commun. Statist. A **17**, 623–640 (1988)

28.37 E.K. Muller, M.C. Mok: The disribution of Cook's D statistics, Commun. Statist. A **26**, 525–546 (1997)

28.38 P.J. Rousseeuw: Least median of squares regression, J. Am. Statist. Assoc. **79**, 871–880 (1984)

28.39 J.W. Wisnowski, D.C. Montgomey, J.R. Simpson: A comparative analysis of multiple outliers detection procedures in the linear regression model, Comput. Statist. Data Anal. **36**, 351–382 (2001)

28.40 D. Peña, J. Rodriguez, G.C. Tiao: Identifying mixtures of regression equations by the SAR procedure (with discussion). In: *Bayesian Statistics*, Vol. 7, ed. by Bernardo et al. (Oxford Univ. Press, Oxford 2003) pp. 327–347

29. Logistic Regression Tree Analysis

This chapter describes a tree-structured extension and generalization of the logistic regression method for fitting models to a binary-valued response variable. The technique overcomes a significant disadvantage of logistic regression viz. the interpretability of the model in the face of multi-collinearity and Simpson's paradox. Section 29.1 summarizes the statistical theory underlying the logistic regression model and the estimation of its parameters. Section 29.2 reviews two standard approaches to model selection for logistic regression, namely, model deviance relative to its degrees of freedom and the Akaike information criterion (AIC) criterion. A dataset on tree damage during a severe thunderstorm is used to compare the approaches and to highlight their weaknesses. A recently published partial one-dimensional model that addresses some of the weaknesses is also reviewed.

Section 29.3 introduces the idea of a logistic regression tree model. The latter consists of a binary tree in which a simple linear logistic regression (i.e., a linear logistic regression using a single predictor variable) is fitted to each leaf node. A split at an intermediate node is characterized by a subset of values taken by a (possibly different) predictor variable. The objective is to partition the dataset into rectangular pieces according to the values of the predictor variables such that a simple linear logistic regression model adequately fits the data in each piece. Because the tree structure and the piecewise models can be presented graphically, the whole model can be easily understood. This is illustrated with the thunderstorm dataset using the LOTUS algorithm.

Section 29.4 describes the basic elements of the LOTUS algorithm, which is based on recursive partitioning and cost-complexity pruning. A key feature of the algorithm is a correction for bias in variable selection at the splits of the tree. Without bias correction, the splits can yield incorrect inferences. Section 29.5 shows an application of LOTUS to a dataset on automobile crash tests involving dummies. This dataset is challenging because of its large size, its mix of ordered and unordered variables, and its large number of missing values. It also provides a demonstration of Simpson's paradox. The chapter concludes with some remarks in Sect. 29.5.

Logistic regression is a technique for modeling the probability of an event in terms of suitable explanatory or predictor variables. For example, [29.1] use it to model the probability that a tree in a forest is blown down during an unusually severe thunderstorm that occurred on July 4, 1999, and caused great damage over 477 000 acres of the Boundary Waters Canoe Area Wilderness in northeastern Minnesota. Data from a sample of 3666 trees were collected, including for each tree, whether it was blown down ($Y = 1$) or not ($Y = 0$), its trunk diameter D in centimeters, its species S, and the local intensity L of the storm, as measured by the fraction of damaged trees in its vicinity. The dataset may be obtained from www.stat.umn.edu/~sandy/pod.

Let $p = \Pr(Y = 1)$ denote the probability that a tree is blown down. In *linear logistic regression*, we model $\log[p/(1-p)]$ as a function of the predictor variables, with the requirement that it be linear in any unknown parameters. The function $\log[p/(1-p)]$ is also often written as $\text{logit}(p)$. If we use a single predictor such

as L, we have the *simple linear* logistic regression model

$$\text{logit}(p) = \log[p/(1-p)] = \beta_0 + \beta_1 L \qquad (29.1)$$

which can be re-expressed in terms of p as $p = \exp(\beta_0 + \beta_1 L)/[1 + \exp(\beta_0 + \beta_1 L)]$.

In general, given k predictor variables $X_1, \ldots, X_k$, a linear logistic regression model in these variables is $\text{logit}(p) = \beta_0 + \sum_{j=1}^{k} \beta_j X_j$. The parameters $\beta_0, \beta_1, \ldots, \beta_k$ are typically estimated using maximum likelihood theory. Let n denote the sample size and let $(x_{i1}, \ldots, x_{ik}, y_i)$ denote the values of $(X_1, \ldots, X_k, Y)$ for the ith observation $(i = 1, \ldots, n)$. Treating each y_i as the outcome of an independent Bernoulli random variable with success probability p_i, we have the likelihood function

$$\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} = \frac{\exp\left[\sum_i y_i \left(\beta_0 + \sum_j \beta_j x_{ij}\right)\right]}{\prod_i \left[1 + \exp\left(\beta_0 + \sum_j \beta_j x_{ij}\right)\right]} .$$

The *maximum likelihood estimates* (MLEs) $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$ are the values of $(\beta_0, \beta_1, \ldots, \beta_k)$ that maximize this function. Newton–Raphson iteration is usually needed to compute the MLEs.

29.1 Approaches to Model Fitting

The result of fitting model (29.1) is

$$\text{logit}(p) = -1.999 + 4.407L. \qquad (29.2)$$

Figure 29.1 shows a plot of the estimated p function. Clearly, the stronger the local storm intensity, the higher the chance for a tree to be blown down.

Figure 29.2 shows boxplots of D by species. Because of the skewness of the distributions, we follow [29.1] and use $\log(D)$, the natural logarithm of D, in our analysis. With $\log(D)$ in place of L, the fitted model becomes

$$\text{logit}(p) = -4.792 + 1.749\log(D) \qquad (29.3)$$

suggesting that tall trees are less likely to survive a storm than short ones. If we use both $\log(D)$ and L, we obtain the model

$$\text{logit}(p) = -6.677 + 1.763\log(D) + 4.420L . \qquad (29.4)$$

Finally, if we include the product $L\log(D)$ to account for interactions between D and L, we obtain

$$\text{logit}(p) = -4.341 + 0.891\log(D) - 1.482L + 2.235L\log(D) . \qquad (29.5)$$

So far, we have ignored the species S of each tree in our sample. We might get a model with higher prediction accuracy if we include S. As with least-squares regression, we can include a categorical variable that takes m distinct values by first defining $m-1$ indicator variables, $U_1, \ldots, U_{m-1}$, each taking the value 0 or 1. The definitions of the indicator variables corresponding

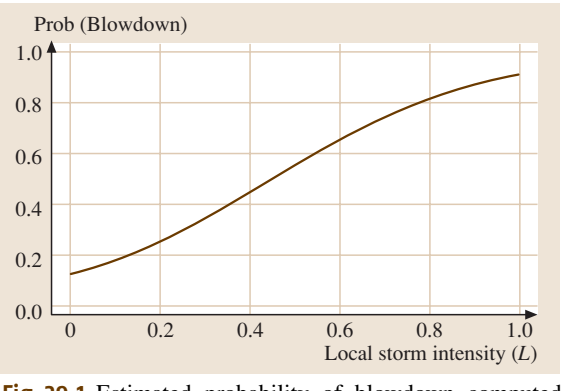


Fig. 29.1 Estimated probability of blowdown computed from a simple linear logistic regression model using L as predictor

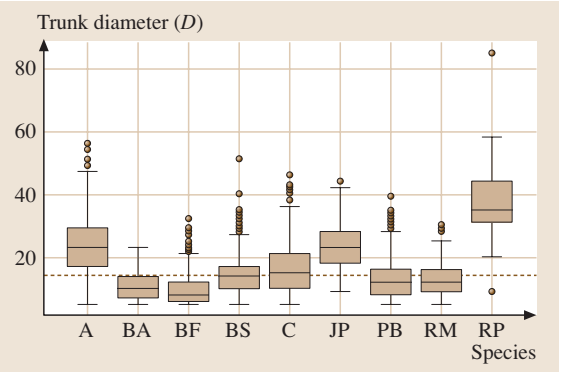


Fig. 29.2 Boxplots of trunk diameter D. The median value of 14 for D, or 2.64 for $\log(D)$, is marked with a *dotted line*

to our nine-species variable S are shown in Table 29.1. Note that we use the *set-to-zero constraint*, which sets all the indicator variables to 0 for the first category (aspen). A model that assumes the same slope coefficients for all species but that gives each a different intercept term is

$$\begin{aligned}\text{logit}(p) = &-5.997 + 1.581\log(D) + 4.629L \\ &- 2.243U_1 + 0.0002U_2 + 0.167U_3 \\ &- 2.077U_4 + 1.040U_5 - 1.724U_6 \\ &- 1.796U_7 - 0.003U_8\,. \end{aligned} \tag{29.6}$$

How well do the models (29.2–29.6) fit the data? One popular method of assessing fit is by means of significance tests based on the model deviance and its *degrees of freedom* (DF)—see, e.g., [29.2] for the definitions. The deviance is analogous to the residual sum of squares in least-squares regression. For the model (29.6), the deviance is 3259 with 3655 DF. We can evaluate the fit of this model by comparing its deviance against that of a larger one, such as the 27-parameter model

$$\begin{aligned}\text{logit}(p) = &\beta_0 + \beta_1\log(D) + \beta_2 L + \sum_{j=1}^{8}\gamma_j U_j \\ &+ \sum_{j=1}^{8}\beta_{1j}U_j\log(D) + \sum_{j=1}^{8}\beta_{2j}U_j L \end{aligned} \tag{29.7}$$

which allows the slope coefficients for $\log(D)$ and L to vary across species. Model (29.7) has a deviance of 3163 with 3639 DF. If the model (29.6) provides a suitable fit to the data, statistical theory says that the difference in deviance should be approximately distributed as a chi-square random variable with DF equal to the difference in the DF of the two models. For our example, the difference in deviance of $3259 - 3163 = 96$ is much too large to be explained by a chi-square distribution with $3655 - 3639 = 16$ DF.

Rejection of model (29.6) does not necessarily imply, however, that the model (29.7) is satisfactory. To find out, we need to compare it with a larger model, such as the 28-parameter model

$$\begin{aligned}\text{logit}(p) = &\beta_0 + \beta_1\log(D) + \beta_2 L + \beta_3 L\log(D) \\ &+ \sum_{j=1}^{8}\gamma_j U_j + \sum_{j=1}^{8}\beta_{1j}U_j\log(D) \\ &+ \sum_{j=1}^{8}\beta_{2j}U_j L \end{aligned} \tag{29.8}$$

which includes an interaction between $\log(D)$ and L. This has a deviance of 3121 with 3638 DF. Model (29.7) is therefore rejected because its deviance differs from that of (29.8) by 42 but their DFs differ only by 1. It turns out that, using this procedure, each of models (29.2–29.7) is rejected when compared against the next larger model in the set.

A second approach chooses a model from a given set by minimizing some criterion that balances model fit with model complexity. One such is the AIC criterion, defined as the deviance plus twice the number of estimated parameters [29.3]. It is well known, however, that the AIC criterion tends to overfit the data. That is, it often chooses a large model. For example, if we apply it to the set of all models up to third order for the current data, it chooses the largest, i. e., the 36-parameter model

$$\begin{aligned}\text{logit}(p) = &\beta_0 + \beta_1\log(D) + \beta_2 L + \sum_{j=1}^{8}\gamma_j U_j \\ &+ \beta_3 L\log(D) + \sum_{j=1}^{8}\beta_{1j}U_j\log(D) \\ &+ \sum_{j=1}^{8}\beta_{2j}U_j L + \sum_{j=1}^{8}\delta_j U_j L\log(D)\,. \end{aligned} \tag{29.9}$$

Table 29.1 Indicator variable coding for the species variable S

Species	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8
A (aspen)	0	0	0	0	0	0	0	0
BA (black ash)	1	0	0	0	0	0	0	0
BF (balsam fir)	0	1	0	0	0	0	0	0
BS (black spruce)	0	0	1	0	0	0	0	0
C (cedar)	0	0	0	1	0	0	0	0
JP (jack pine)	0	0	0	0	1	0	0	0
PB (paper birch)	0	0	0	0	0	1	0	0
RM (red maple)	0	0	0	0	0	0	1	0
RP (red pine)	0	0	0	0	0	0	0	1

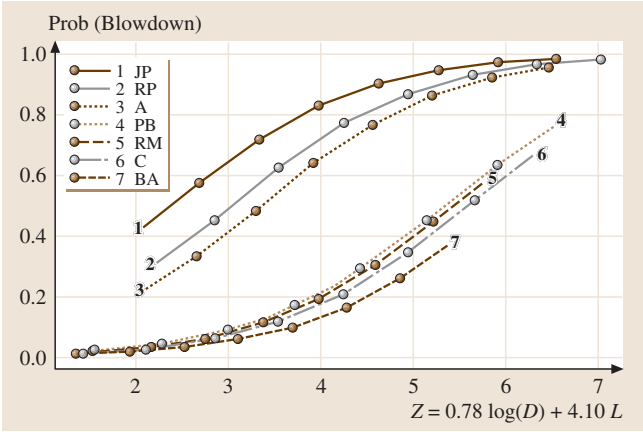


Fig. 29.3 Estimated probability of blowdown for seven speciesTable 29.1, excluding balsam fir (BF) and black spruce (BS), according to model (29.10)

Graphical interpretation of models (29.8) and (29.9) is impossible. The simple and intuitive solution of viewing the estimated p-function by a graph such as Fig. 29.1 is unavailable when a model involves more than one predictor variable. This problem is exacerbated by the fact that model complexity typically increases with increasing sample size or number of predictors. Interpretation of the estimated coefficients is frequently futile, because the estimates typically do not remain the same from one model to another. For example, the coefficient for L is 4.407, 4.424, 1.870, and 4.632 in models (29.2), (29.4), (29.5), and (29.6), respectively. This is due to multi-collinearity among the predictor variables.

Cook and *Weisberg* [29.1] try to solve the problem of interpretation by using a *partial one-dimensional* (POD) model, which employs a single linear combination of the noncategorical variables, $Z = \delta_1 \log(D) + \delta_2 L$, as predictor. For the tree data, they find that if balsam fir (BF) and black spruce (BS) are excluded, the model $\text{logit}(p) = \beta_0 + Z + \sum_j \gamma_j U_j$, with $Z = 0.78 \log(D) + 4.1L$, fits the other species quite well. One advantage of this model is that the estimated p-functions may be displayed graphically, as shown in Fig. 29.3. The graph is not as natural as Fig. 29.1, however, because Z is a linear combination of two variables. In order to include the species BF and BS, [29.1] choose the larger model

$$\begin{aligned}\text{logit}(p) = \beta_0 + Z + \sum_{j=1}^{9} \gamma_j U_j \\ + (\theta_1 I_{\text{BF}} + \theta_2 I_{\text{BS}}) \log(D) \\ + (\phi_1 I_{\text{BF}} + \phi_2 I_{\text{BS}}) L \end{aligned} \quad (29.10)$$

which contains separate coefficients, θ_j and ϕ_j, for BF and BS. Here $I_{(\cdot)}$ denotes the indicator function, i. e., $I_A = 1$ if the species is A, and $I_A = 0$ otherwise. Of course, this model does not allow a graphical representation for BF and BS.

29.2 Logistic Regression Trees

The type of models and the method of selection described in the previous section are clearly not totally satisfactory. As the sample size or the number of predictor variables increases, so typically does model complexity. But a more complex model is always harder to interpret than a simple one. On the other hand, an overly simple model may have little predictive power.

A logistic regression tree model offers one way to retain simultaneously the graphical interpretability of simple models and the predictive accuracy of richer ones. Its underlying motivation is that of divide and conquer. That is, a complex set of data is divided into sufficiently many subsets such that a simple linear logistic regression model adequately fits the data in each subset. Data subsetting is performed recursively, with the sample split on one variable at a time. This results in the partitions being representable as a binary decision tree. The method is implemented by [29.4] in a computer program called LOTUS.

Figure 29.4 shows a LOTUS model fitted to the tree data. The data are divided into ten subsets, labeled 0–9. Balsam fir (BF), one of the two species excluded from the [29.1] model, is isolated in subsets 0 and 9, where $\log(D)$ is the best linear predictor. The estimated p-functions for these two subsets are shown in Fig. 29.5. The estimated p-functions for the trees that are not balsam firs can be displayed together in one graph, as shown in Fig. 29.6, because they all employ L as the best simple linear predictor.

From the graphs, we can draw the following conclusions:

1. The probability of blowdown consistently increases with D and L, although the value and rate of increase are species-dependent.

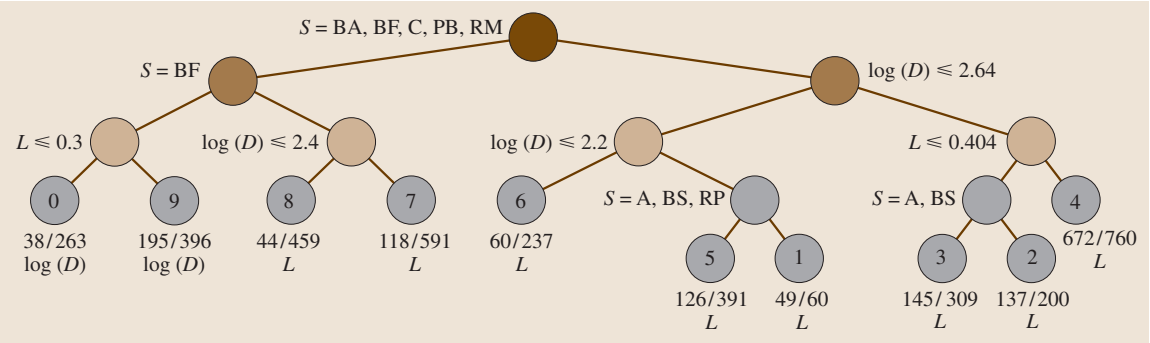


Fig. 29.4 A piecewise simple linear LOTUS model for estimating the probability that a tree is blown down. A splitting rule is given beside each intermediate node. If a case satisfies the rule, it goes to the left child node; otherwise the right child node. The second level split at $\log(D) = 2.64$ corresponds to the median value of D. Beneath each leaf node are the ratio of cases with $Y = 1$ to the node sample size and the name of the selected predictor variable

2. Balsam fir (BF) has the highest chance of blowdown, given any values of D and L.
3. The eight species excluding the balsam fir can be divided into two groups. Group I consists of black ash (BA), cedar (C), paper birch (PB), and red maple (RM). They belong to subsets 7 and 8, and are most likely to survive. This is consistent with the POD model of [29.1]. Group II contains aspen (A), black spruce (BS), jack pine (JP), and red pine (RP).
4. The closeness of the estimated p-functions for subsets 6 and 7 show that the smaller group II trees and the larger group I trees have similar blowdown probabilities for any given value of L.
5. Although aspen (A) and black spruce (BS) are always grouped together, namely, in subsets 3–6, less than 15% of the aspen trees are in subsets 5 and 6. Similarly, only 2% of the red pines (RP) are in these two subsets. Hence the p-function of aspen (A) is mainly described by the curves for subsets 3 and 4, and that for red pine (RP) by the curves for

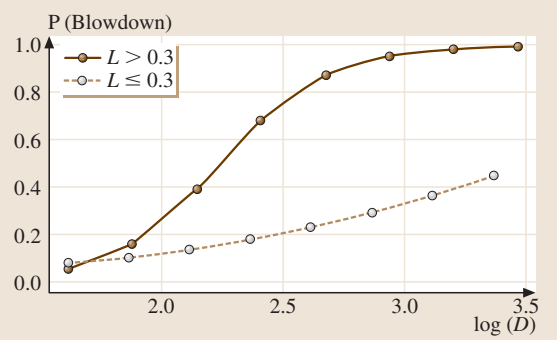


Fig. 29.5 Estimated probability of blowdown for balsam fir (BF), according to the LOTUS model in Fig. 29.4

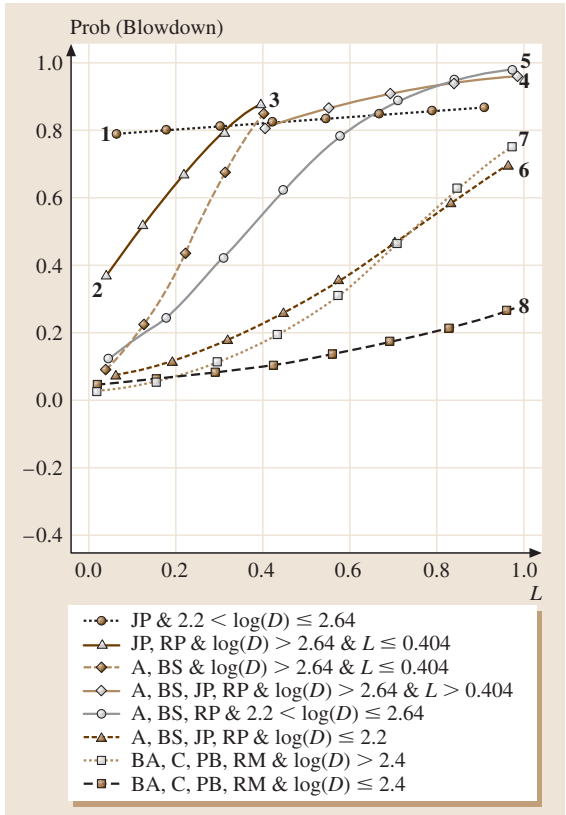


Fig. 29.6 Estimated probability of blowdown for all species except balsam firs, according to the LOTUS model in Fig. 29.4

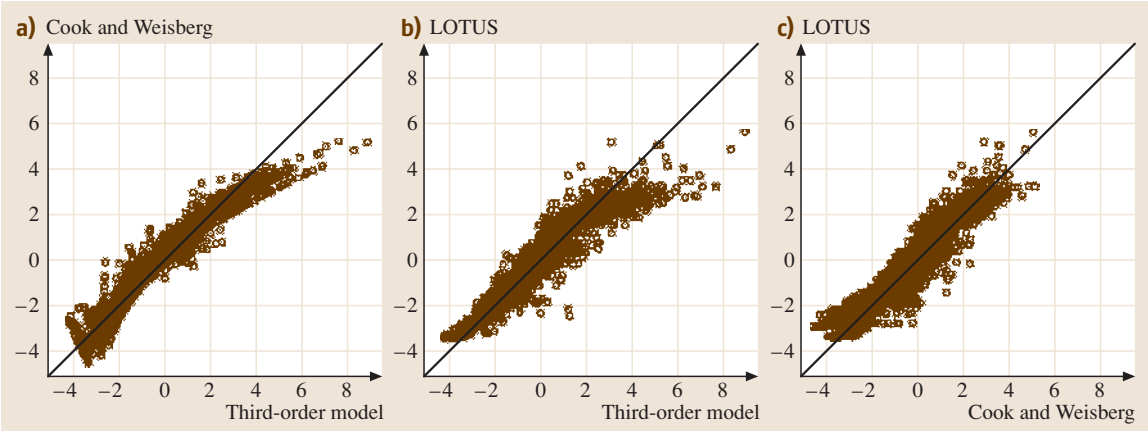


Fig. 29.7a–c Comparison of fitted logit values among three models. **(a)** Cook & Weisenberg versus third-order model **(b)** LOTUS versus third-order model **(c)** Cook & Weisenberg versus LOTUS

subsets 2 and 4. We conclude that, after balsam fir (BF), the three species most at risk of blowdown are the jack pine (JP), red pine (RP), and aspen (A), in that order. This ordering of JP, RP, and A is the same as the POD model of [29.1], as can be seen in Fig. 29.3.

6. Recall that the black spruce (BS) was the other species that [29.1] could not include in their POD model. The reason for this is quite clear from Fig. 29.6, where we use solid lines to draw the estimated p-function for black spruce. Four curves are required, corresponding to subsets 3, 4, 5, and 6. The spread of these curves suggests that the p-function of black spruce is highly sensitive to changes in D. This explains why the species cannot be included in the POD model.

How does the LOTUS model compare with the others? The former is clearly superior in terms of interpretability. But does it predict future observations as well as the other models? Unfortunately, this question cannot be answered completely, because we do not have an independent set of data to test the models. The best we can do is to compare the fitted values from the different models. This is done in Fig. 29.7, which plots the fitted logit values of the LOTUS model against those of the POD and the linear logistic regression model with all interactions up to third order. The plots show that there is not much to choose among them.

29.3 LOTUS Algorithm

As already mentioned, the idea behind LOTUS is to partition the sample space into one or more pieces such that a simple model can be fitted to each piece. This raises two issues: (i) how to carry out the partitioning, and (ii) how to decide when a partition is good enough. We discuss each question in turn.

29.3.1 Recursive Partitioning

Like all other regression tree algorithms, LOTUS splits the dataset recursively, each time choosing a single variable X for the split. If X is an ordered variable, the split has the form $s = \{X \le c\}$, where c is a constant. On the other hand, if X is a categorical variable, the split has the form $s = \{X \in \omega\}$, where ω is a subset of the values taken by X. The way s is chosen is critically important if the tree structure is to be used for inference about the variables.

For least-squares regression trees, many algorithms, such as automatic interaction detector (AID) [29.5], CART [29.6] and M5 [29.7], choose the split s that minimizes the total sum of squared residuals of the regression models fitted to the two data subsets created by s. Although this approach can be directly extended to logistic regression by replacing the sum of squared residuals with the deviance, it is fundamentally flawed, because it is biased toward choosing X

variables that allow more splits. To see this, suppose that X is an ordered variable taking n unique values. Then there are $n-1$ ways to split the data along the X axis, with each split $s = \{X \le c\}$ being such that c is the midpoint between two consecutively ordered values. This creates a selection bias toward X variables for which n is large. For example, in our tree dataset, variable L has 709 unique values but variable $\log(D)$ has only 87. Hence if all other things are equal, L is eight times more likely to be selected than $\log(D)$.

The situation can be worse if there are one or more categorical X variables with many values. If X takes n categorical values, there are $2^{n-1}-1$ splits of the form $s = \{X \in \omega\}$. Thus the number of splits grows exponentially with the number of categorical values. In our example, the species variable S generates $2^{9-1}-1 = 255$ splits, almost three times as many splits as $\log(D)$.

Doyle [29.8] was the first to warn that this bias can yield incorrect inferences about the effects of the variables. The GUIDE [29.9] least-squares regression tree algorithm avoids the bias by employing a two-step approach to split selection. First, it uses statistical significance tests to select X. Then it searches for c or ω. The default behavior of GUIDE is to use categorical variables for split selection only; they are not converted into indicator variables for regression modeling in the nodes. LOTUS extends this approach to logistic regression. The details are given in [29.4], but the essential steps in the recursive partitioning algorithm can be described as follows.

1. Fit a logistic regression model to the data using only the noncategorical variables.
2. For each ordered X variable, discretize its values into five groups at the sample quintiles. Form a 2×5 contingency table with the Y values as rows and the five X groups as columns. Compute the significance probability of a trend-adjusted chi-square test for nonlinearity in the data.
3. For each categorical X variable, since they are not used as predictors in the logistic regression models, compute the significance probability of the chi-square test of association between Y and X.
4. Select the variable with the smallest significance probability to partition the data.

By using tests of statistical significance, the selection-bias problem due to some X variables taking more values than others disappears. Simulation results to support the claim are given in [29.4].

After the X variable is selected, the split value c or split subset ω can be found in many ways. At the time of this writing, LOTUS examines only five candidates. If X is an ordered variable, LOTUS evaluates the splits at c equal to the 0.3, 0.4, 0.5, 0.6, and 0.7 quantiles of X. If X is categorical, it evaluates the five splits around the subset ω that minimizes a weighted sum of the binomial variance in Y in the two partitions induced by the split. The full details are given in [29.4]. For each candidate split, LOTUS computes the sum of the deviances in the logistic regression models fitted to the data subsets. The split with the smallest sum of deviances is selected.

29.3.2 Tree Selection

Instead of trying to decide when to stop the partitioning, GUIDE and LOTUS follow the CART method of first growing a very big tree and then progressively pruning it back to the root node. This yields a nested sequence of trees from which one is chosen. If an independent test dataset is available, the choice is easy: just apply each tree in the sequence to the test set and choose the tree with the lowest prediction deviance.

If a test set is not available, as is the case in our example, the choice is made by ten-fold crossvalidation. The original dataset is divided randomly into ten subsets. Leaving out one subset at a time, the entire tree-growing process is applied to the data in the remaining nine subsets to obtain another nested sequence of trees. The subset that is left out is then used as a test set to evaluate this sequence. After the process is repeated ten times, by leaving out one subset in turn each time, the combined results are used to choose a tree from the original tree sequence grown from all the data. The reader is referred to [29.6, Chapt. 3] for details on pruning and tree selection. The only difference between CART and LOTUS here is that LOTUS uses deviance instead of the sum of squared residuals.

29.4 Example with Missing Values

We now show how LOTUS works when the dataset has missing values. We use a large dataset from the National Highway Transportation Safety Administration (ftp://www.nhtsa.dot.gov/ges) on crash tests of vehicles

Table 29.2 Predictor variables in the crash-test dataset. Angular variables `crbang`, `pdof`, and `impang` are measured in degrees clockwise (from -179 to 180) with 0 being straight ahead

Name	Description	Variable type
make	Vehicle manufacturer	63 categories
model	Vehicle model	466 categories
year	Vehicle model year	continuous
body	Vehicle body type	18 categories
engine	Engine type	15 categories
engdsp	Engine displacement	continuous
transm	Transmission type	7 categories
vehtwt	Vehicle test weight	continuous
vehwid	Vehicle width	continuous
colmec	Steering column collapse mechanism	10 categories
modind	Vehicle modification indicator	4 categories
vehspd	Resultant speed of vehicle before impact	continuous
crbang	Crabbed angle	continuous
pdof	Principal direction of force	continuous
tksurf	Test track surface	5 categories
tkcond	Test track condition	6 categories
impang	Impact angle	continuous
occloc	Occupant location	6 categories
occtyp	Occupant type	12 categories
dumsiz	Dummy size percentile	8 categories
seposn	Seat position	6 categories
rsttyp	Restraint type	26 categories
barrig	Rigid or deformable barrier	2 categories
barshp	Barrier shape	15 categories

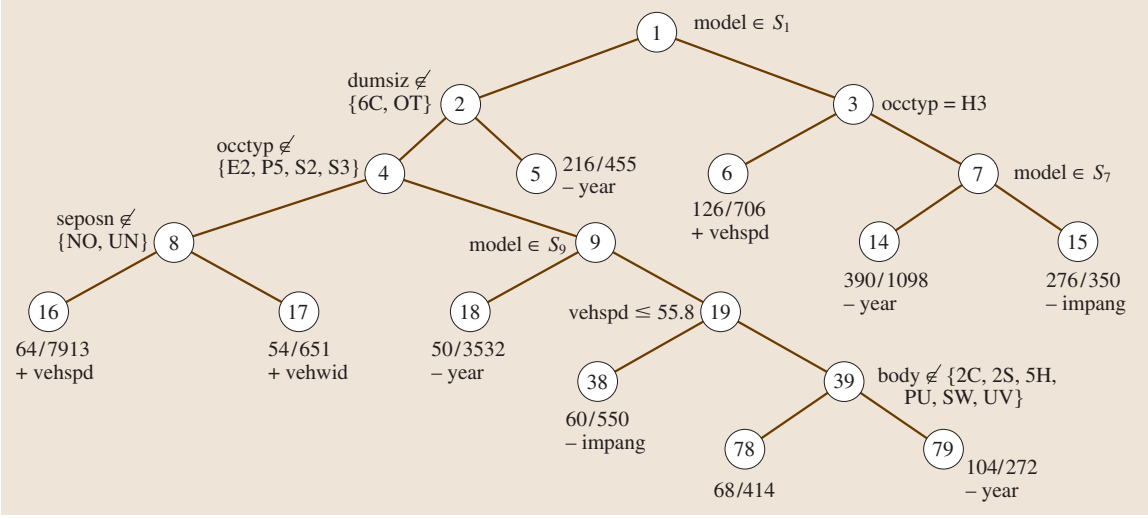


Fig. 29.8 LOTUS model for the crash-test data. Next to each leaf node is a fraction showing the number of cases with $Y = 1$ divided by the sample size, and the name of the best predictor variable, provided it is statistically significant. If the latter has a positive regression coefficient, a plus sign is attached to its name; otherwise a minus sign is shown. The constituents of the sets S_1, S_7, and S_9 may be found from Tables 29.3 and 29.4

involving test dummies. The dataset gives the results of 15 941 crash tests conducted between 1972 and 2004. Each record consists of measurements from the crash of a vehicle into a fixed barrier. The head injury criterion (`hic`), which is the amount of head injury sustained by a test dummy seated in the vehicle, is the main variable of interest. Also reported are eight continuous variables and 16 categorical variables; their names and descriptions are given in Table 29.2. For our purposes, we define $Y = 1$ if `hic` exceeds 1000, and $Y = 0$ otherwise. Thus Y indicates when severe head injury occurs.

One thousand two hundred and eleven of the records are missing one or more data values. Therefore a linear logistic regression using all the variables can be fitted only to the subset of 14 730 records that have complete values. After transforming each categorical variable into a set of indicator variables, the model has 561 regression coefficients, including the constant term. All but six variables (`engine`, `vehwid`, `tkcond`, `impang`, `rsttyp`, and `barrig`) are statistically significant. As mentioned in Sect. 29.1, however, the regression coefficients in the model cannot be relied upon to explain how each variable affects $p = P(Y = 1)$. For example, although `vehspd` is highly significant in this model, it is not significant in a simple linear logistic model that employs it as the only predictor. This phenomenon is known as Simpson's paradox. It occurs when a variable has an effect in the same direction within subsets of the data, but when the subsets are combined, the effect vanishes or reverses in direction.

Being composed of piecewise simple linear logistic models, LOTUS is quite resistant to Simpson's paradox. Further, by partitioning the dataset one variable at a time, LOTUS can use all the information in the dataset, instead of only the complete data records. Specifically, when LOTUS fits a simple linear logistic model to a data subset, it uses all the records that have complete information in Y and the X variable used in the model. Similarly, when X is being evaluated for split selection, the chi-square test is applied to all the records in the subset that have complete information in X and Y.

Figure 29.8 gives the LOTUS tree fitted to the crash-test data. The splits together with the p-functions fitted to the leaf nodes in Fig. 29.9 yield the following conclusions:

1. The tree splits first on `model`, showing that there are significant differences, with respect to p, among vehicle models. The variable is also selected for splitting in nodes 7 and 9. Tables 29.3 and 29.4 give the precise nature of the splits.
2. Immediately below the root node, the tree splits on `dumsiz` and `occtyp`, two characteristics of the test dummy. This shows that some types of dummies are more susceptible to severe injury than others. In

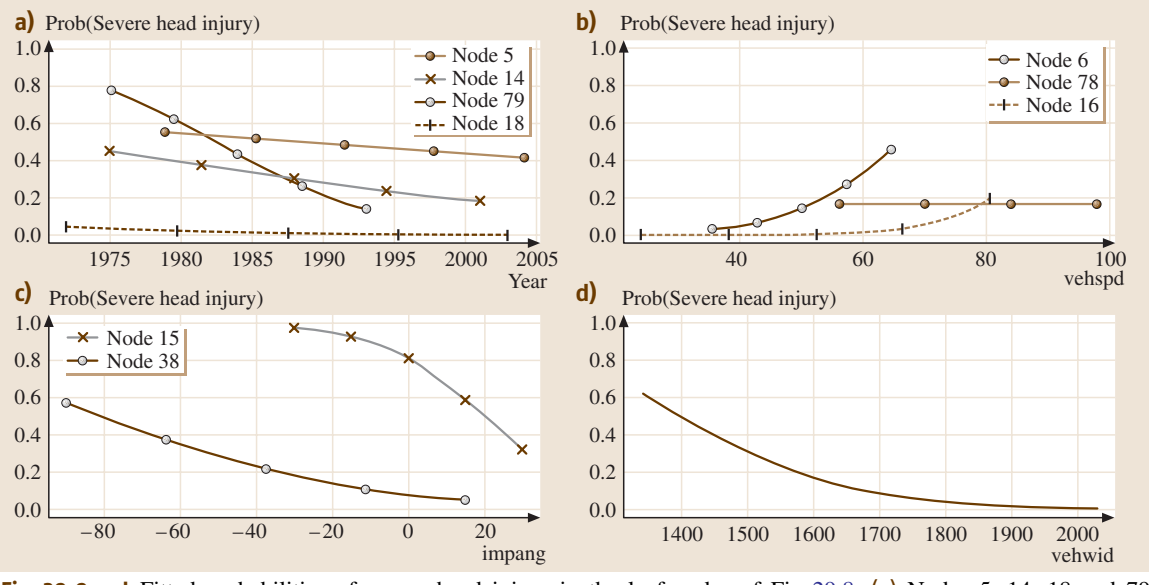


Fig. 29.9a–d Fitted probabilities of severe head injury in the leaf nodes of Fig. 29.8. **(a)** Nodes 5, 14, 18 and 79 **(b)** Nodes 6, 16, and 78 **(c)** Nodes 15 and 38 **(d)** Node 17

Table 29.3 Split at node 7 of the tree in Fig. 29.8

Make	Node 14	Node 15
American	Concord	
Audi	4000, 5000	
Buick	Electra	
Champion	Motorhome	
Chevrolet	K20 Pickup, Monza, Nova, S10 Blazer, Spectrum, Sportvan	Astro, Malibu, Sprint
Chrysler	Imperial, Lebaron	Intrepid
Comuta-Car	Electric	
Dodge	Aries, Challenger, Colt, Lancer, Magnum	Colt Pickup, St. Regis
Ford	Clubwagon MPV, Courier, E100 Van, EXP, Fairmont, Fiesta, Granada, Merkur	Torino
GMC	Sportvan	
Hyundai	Excel GLS	
Isuzu	Impulse, Spacecab	I-Mark, Trooper II
Jeep	Comanche	
Kia	Sorento	
Lectric	Leopard	
Mazda	GLC	B2000
Mercury		Bobcat
Mitsubishi	Montero, Tredia	Pickup
Nissan	2000, 210, Kingcab Pickup, Murano	
Oldsmobile		98
Peugeot		504, 505
Plymouth	Champ, Fury, Horizon	Breeze, Volare
Pontiac	T1000	
Renault	18, Alliance, LeCar, Medallion	Fuego, Sportswagon
Saab	38235	
Saturn	L200	
Subaru	GF, GLF, Wagon	
Suzuki	Sidekick	
Toyota	Celica, Starlet	
Volkswagen	Fox, Scirocco	Beetle, EuroVan
Volvo	244, XC90	
Yugo	GV	

particular, the cases in node 5 contain mainly dummies that correspond to a six-year-old child. The fitted p-function for this node can be seen in the upper left panel of Fig. 29.9. Compared with the fitted p-functions of the other nodes, this node appears to have among the highest values of p. This suggests that six-year-old children are most at risk of injury. They may be too big for child car seats and too small for adult seat belts.

3. The split on `seposn` at node 8 shows that passengers in vehicles with adjustable seats are ten times (average p of 0.008 versus 0.08) less likely to suffer severe head injury than those with nonadjustable seats. This could be due to the former type of vehicle being more expensive and hence able to withstand collisions better.
4. Similarly, the split on `body` at node 39 shows that passengers in two-door cars, pick-ups, station wagons, and sports utility vehicles (SUVs) are twice as likely (average p of 0.38 versus 0.16) to suffer severe head injury than other vehicles.
5. The linear predictor variables selected in each leaf node tell us the behavior of the p-function within each partition of the dataset. Four nodes have `year` as their best linear predictor. Their fitted p-functions are shown in the upper left panel of Fig. 29.9. The

Table 29.4 Split at node 9 of the tree in Fig. 29.8

Make	Node 18	Node 19
Acura	Integra, Legend, Vigor	2.5TL, 3.2TL, 3.5RL, MDX, RSX
American	Gremlin, Matador, Spirit	
Audi	100, 200, 80	A4, A6, A8
Battronics	Van	
BMW	325I, 525I	318, 328I, X5, Z4 Roadster
Buick	Century, LeSabre, Regal, Riviera, Skyhawk, Skylark, Somerset	ParkAvenue, Rendezvous, Roadmaster
Cadillac	Deville, Seville	Brougham, Catera, Concourse, CTS, Eldorado, Fleetwood
Chevrolet	Beretta, Camaro, Cavalier, Celebrity, Chevette, Citation, Corsica, Corvette, Elcamino, Impala, Lumina, LUV, MonteCarlo, Pickup, S-10, Vega	Avalanche, Beauville, Blazer, C-1500, K2500 Pickup, Silverado, Suburban, Tahoe, Tracker, Trailblazer, Venture,
Chinook	Motorhome	
Chrysler	Cirrus, Conquest, FifthAvenue, Newport, NewYorker	LHS, Pacifica, PT Cruiser, Sebring Convertible
Daewoo		Leganza, Nubira
Daihatsu	Charade	
Delorean	Coupe	
Dodge	400, 600, Caravan, D-150, Dakota, Daytona, Diplomat, Dynasty, Mirada, Neon, Rampage, Ramwagonvan, Sportsman	Avenger, Durango, Grand Caravan, Intrepid, Omni, Ram150, Ram1500, Ram, Ram250 Van, Shadow, Spirit, Stratus
Eagle	Medallion, MPV, Premier	Summit, Vision
Eva	Evcort	
Fiat	131, Strada	
Ford	Bronco, Bronco II, Crown Victoria, Escort, F150 Pickup, F250 Pickup, F350 Pickup, Festiva, LTD, Mustang, Pickup, Probe, Ranger, Taurus, Thunderbird, Van, Windstar	Aerostar, Aspire, Contour, E150 Van, Escape, Escort ZX2, EV Ranger, Expedition, Explorer, Focus, Freestar, Other, Tempo
Geo	Metro, Prizm	Storm, Tracker
GMC	Astro Truck, Vandura	EV1
Holden		Commodore Acclaim
Honda	Accord	Civic, CRV, Element, Insight, Odyssey, Pilot, Prelude, S2000
Hyundai	Elantra, Scoupe, Sonata	Accent, Pony Excel, Santa Fe, Tiburon
IH	Scout MPV	
Infinity	G20, M30	J30
Isuzu	Amigo, Pup	Axiom, Pickup, Rodeo, Stylus
Jaguar		X-Type
Jeep	CJ, Wrangler	Cherokee, Cherokee Laredo, Grand Cherokee, Liberty
Jet	Courier, Electrica, Electrica 007	
Kia	Sephia	Rio, Sedona, Spectra, Sportage

Table 29.5 Split at node 9 of the tree in Fig. 29.8 (cont.)

Make	Node 18	Node 19
Landrover		Discovery, Discovery II
Lectra	400, Centauri	
Lexus	ES250	ES300, GS300, GS400, IS300, RX300, RX330
Lincoln	Continental, Town Car	LS, Mark, Navigator
Mazda	323, 323-Protege, 929, Miata, Millenia, MPV, MX3, MX6, Pickup, RX	626, Mazda6, MX5
Mercedes	190, 240, 300	C220, C230, C240, E320, ML320
Mercury	Capri, Cougar, Lynx, Marquis, Monarch, Sable, Topaz, Tracer, Villager, Zephyr	Mystique
Mini		Cooper
Mitsubishi	Diamante, Eclipse, Galant, Mightymax, Mirage, Precis, Starion, Van	3000GT, Cordia, Endeavor, Lancer, Montero Sport, Outlander
Nissan	240SX, 810, Altima, Axxess, Pathfinder, Pulsar, Quest, Sentra, Van	200SX, 300ZX, 350Z, Frontier, Maxima, Pickup, Stanza, Xterra
Odyssey	Motorhome	
Oldsmobile	Calais, Cutlass, Delta 88, Omega, Toronado	Achieva, Aurora, Intrigue, Royale
Other	Other	
Peugeot	604	
Plymouth	Acclaim, Caravelle, Laser, Reliant, Sundance, Voyager	Colt Vista, Conquest, Neon
Pontiac	Bonneville, Fiero, Firebird, Grand AM, Lemans, Parisienne, Sunbird	Aztek, Grand Prix, Sunfire, Trans Sport
Renaissance		Tropica
Renault	Encore	
Saab	900	38233, 9000
Saturn	SL1	Ion, LS, LS2, SC1, SL2, Vue
Sebring		ZEV
Solectria		E-10, Force
Subaru	DL, Impreza, Justy, XT	Forestee, GL, Legacy
Suzuki	Samurai	Swift, Vitara
Toyota	Camry, Corolla, Corona, Cosmo, Landcruiser, MR2, Paseo, T100, Tercel, Van	4Runner, Avalon, Camry Solara, Cressida, Echo, Highlander, Matrix, Pickup, Previa, Prius, Rav4, Sequoia, Sienna, Tacoma, Tundra
UM	Electrek	
Volkswagen	Cabrio, Corrado, Golf, Passat, Quantum, Rabbit	Jetta, Polo, Vanagon
Volvo	240, 740GL, 850, 940, DL, GLE	960, S60, S70, S80
Winnebago	Trekker	

decreasing trends show that crash safety is improving over time.

6. Three nodes have `vehspd` as their best linear predictor, although the variable is not statistically significant in one (node 78). The fitted p-functions are shown in the upper right panel of Fig. 29.9. As expected, p is nondecreasing in `vehspd`.
7. Two nodes employ `impang` as their best linear predictor. The fitted p-functions shown in the bottom left panel of Fig. 29.9 suggest that side impacts are more serious than frontal impacts.

8. One node has `vehwid` as its best linear predictor. The decreasing fitted p-function shown in the lower right panel of Fig. 29.9 shows that vehicles that are smaller are less safe.

29.5 Conclusion

Logistic regression is a statistical technique for modeling the probability p of an event in terms of the values of one or more predictor variables. The traditional approach expresses the logit of p as a linear function of these variables. Although the model can be effective for predicting p, it is notoriously hard to interpret. In particular, multi-collinearity can cause the regression coefficients to be misinterpreted.

A logistic regression tree model offers a practical alternative. The model has two components, namely, a binary tree structure showing the data partitions and a set of simple linear logistic models, fitted one to each partition. It is this division of model complexity that makes the model intuitive to interpret. By dividing the dataset into several pieces, the sample space is effectively split into different strata such that the p-function is adequately explained by a single predictor variable in each stratum. This property is powerful because: (i) the partitions can be understood through the binary tree, and (ii) each p-function can be visualized through its own graph. Further, stratification renders each of the individual p-functions resistant to the ravages of multi-collinearity among the predictor variables and to Simpson's paradox. Despite these advantages, it is crucial for the partitioning algorithm to be free of selection bias. Otherwise, it is very easy to draw misleading inferences from the tree structure. At the time of writing, LOTUS is the only logistic regression tree algorithm designed to control such bias.

Finally, as a disclaimer, it is important to remember that, in real applications, there is no *best* model for a given dataset. This situation is not unique to logistic regression problems; it is prevalent in least-squares and other forms of regression as well. Often there are two or more models that give predictions of comparable average accuracy. Thus a LOTUS model should be regarded as merely one of several possibly different ways of explaining the data. Its main virtue is that, unlike many other methods, it provides an *interpretable* explanation.

References

29.1 R. D. Cook, S. Weisberg: Partial one-dimensional regression models, Am. Stat. **58**, 110–116 (2004)

29.2 A. Agresti: *An Introduction to Categorical Data Analysis* (Wiley, New York 1996)

29.3 J. M. Chambers, T. J. Hastie: *Statistical Models in S* (Wadsworth, Pacific Grove 1992)

29.4 K.-Y. Chan, W.-Y. Loh: LOTUS: An algorithm for building accurate and comprehensible logistic regression trees, J. Comp. Graph. Stat. **13**, 826–852 (2004)

29.5 J. N. Morgan, J. A. Sonquist: Problems in the analysis of survey data, and a proposal, J. Am. Stat. Assoc. **58**, 415–434 (1963)

29.6 L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone: *Classification and Regression Trees* (Wadsworth, Belmont 1984)

29.7 J. R. Quinlan: *Learning with continuous classes, Proceedings of AI'92 Australian National Conference on Artificial Intelligence* (World Scientific, Singapore 1992) pp. 343–348

29.8 P. Doyle: The use of automatic interaction detector and similar search procedures, Oper. Res. Q. **24**, 465–467 (1973)

29.9 W.-Y. Loh: Regression trees with unbiased variable selection and interaction detection, Stat. Sin. **12**, 361–386 (2002)

30. Tree-Based Methods and Their Applications

The first part of this chapter introduces the basic structure of tree-based methods using two examples. First, a classification tree is presented that uses e-mail text characteristics to identify spam. The second example uses a regression tree to estimate structural costs for seismic rehabilitation of various types of buildings. Our main focus in this section is the interpretive value of the resulting models.

This brief introduction is followed by a more detailed look at how these tree models are constructed. In the second section, we describe the algorithm employed by classification and regression tree (CART), a popular commercial software program for constructing trees for both classification and regression problems. In each case, we outline the processes of growing and pruning trees and discuss available options. The section concludes with a discussion of practical issues, including estimating a tree's predictive ability, handling missing data, assessing variable importance, and considering the effects of changes to the learning sample.

The third section presents several alternatives to the algorithms used by CART. We begin with a look at one class of algorithms – including QUEST, CRUISE, and GUIDE– which is designed to reduce potential bias toward variables with large numbers of available splitting values. Next, we explore C4.5, another program popular in the artificial-intelligence and machine-learning communities. C4.5 offers the added functionality of converting any tree to a series of *decision rules*, providing an alternative means of viewing and interpreting its results. Finally, we discuss chi-square automatic interaction detection (CHAID), an early classification-tree construction algorithm used with categorical predictors. The section concludes with a brief comparison of the characteristics of CART and each of these alternative algorithms.

In the fourth section, we discuss the use of ensemble methods for improving predictive ability. Ensemble methods generate collections of trees using different subsets of the training data. Final predictions are obtained by aggregating over the predictions of individual members of these collections. The first ensemble method we consider is boosting, a recursive method of generating small trees that each specialize in predicting cases for which its predecessors perform poorly. Next, we explore the use of random forests, which generate collections of trees based on bootstrap sampling procedures. We also comment on the tradeoff between the predictive power of ensemble methods and the interpretive value of their single-tree counterparts.

The chapter concludes with a discussion of tree-based methods in the broader context of supervised learning techniques. In particular, we compare classification and regression trees to multivariate adaptive regression splines, neural networks, and support vector machines.

30.1 Overview

Given a data set for a particular application, a researcher will typically build a statistical model with one (or both) of the following objectives in mind: (1) to use information from this data to make useful predictions about future observations, and (2) to gain some insights into the underlying structure of the data. Tree-based models are attractive because of their potential to blend both of these characteristics quite effectively.

Tree-based models comprise one set of tools useful for supervised learning tasks. In supervised learning problems, a researcher is trying to use a set of *inputs*, or *independent variables*, to predict an *output*, or *dependent variable*. If the output is a categorical variable, we call this a problem of classification. On the other hand, if the output is a continuous variable, we call this a problem of regression.

Tree-based models approach these problems by recursively partitioning a learning sample over its input variable space and fitting a simple function to each resulting subgroup of cases. In classification, this function is assignment to a single category; in regression, the function could be a constant. We shall discuss several tree-fitting procedures in detail throughout this chapter.

To see the tree-based models at work, we present two applications in this section.

30.1.1 Classification Example: Spam Filtering

First, we consider the task of designing an automatic spam (junk e-mail) filter. The data for this task are publicly available from the University of California, Irvine (UCI) machine learning repository [30.1], and were donated by George Forman from Hewlett–Packard laboratories in Palo Alto, California.

The data consist of 58 variables describing 4601 messages. The dependent variable indicates whether or not each message is spam. The 57 predictor variables are all continuous, and describe the relative frequencies of various keywords, characters, and strings of consecutive uppercase letters. A resulting tree model is shown in Fig. 30.1, and the variables present in the tree are summarized in Table 30.1.

In Fig. 30.1, we see that the messages are first partitioned based on the frequency of the "$" character. Messages with few dollar signs are sent down the left branch, and messages with many dollar signs are sent down the right branch. Following the right branch, we find that those messages with many dollar signs are further partitioned based on the frequency of the word "hp". If "hp" appears only infrequently, the message is classified as spam, otherwise, it is classified as a legitimate message. This splitting makes sense in the context of this data set, because messages containing "hp" most likely address company business.

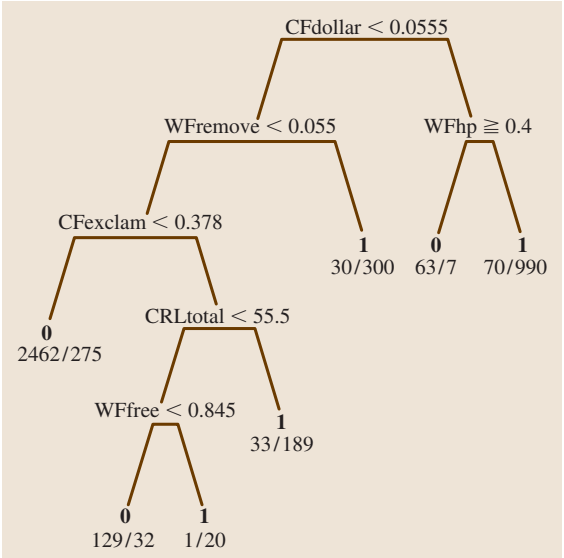


Fig. 30.1 A classification tree for the spam filtering data. Terminal nodes labeled "1" are classified as spam, and those labeled "0" are classified as non-spam

Following the left branch from the root (top) node, we find that a message will be classified as spam if it has a high frequency of the word "remove", or a combination of exclamation points and either the word "free" or many strings of uppercase letters. The structure of the tree is consistent with our intuition about the message characteristics that separate spam from legitimate e-mail.

Table 30.1 Electronic mail characteristics

Variable	Definition
spam	1 if spam, 0 if not
CFdollar	percent of "$" characters in message
CFexclam	percent of "!" characters in message
CRLtotal	sum of lengths of uppercase letter strings
WFfree	percent of "free" words in message
WFhp	percent of "hp" words in message
WFremove	percent of "remove" words in message

Table 30.2 Seismic rehabilitation cost-estimator variables

Variable	Definition
AGE	Year of construction
AREA	Building area in square feet
MODELC1	Building has concrete moment frame (yes = 1, no = 0)
MODELC3	Building has concrete frame w/ infill walls (yes = 1, no = 0)
MODELS5	Building has steel frame w/ infill walls (yes = 1, no = 0)
MODELURM	Building has unreinforced masonry (yes = 1, no = 0)
MODELW1	Building has light wood frame (yes = 1, no = 0)
POBJ_DC	Performance objective is damage control (yes = 1, no = 0)
POBJ_IO	Performance objective is immediate occupancy (yes = 1, no = 0)
POBJ_RR	Performance objective is risk reduction (yes = 1, no = 0)
SEISMIC	Location seismicity on a scale from 1 (low) to 7 (very high)

We have chosen a small tree for the sake of illustration. For this particular application, one may consider competing methods such as logistic regression or logistic regression trees described in Chapt. 29 by Loh; also see *Chan* and *Loh* [30.2]. Having seen a successful classification example, we now examine a regression tree application.

30.1.2 Regression Example: Seismic Rehabilitation Cost Estimator

The seismic rehabilitation cost estimator is an online program developed by the Federal Emergency Management Agency (FEMA) [30.3,4] that enables calculation of structural cost estimates for seismic rehabilitation of buildings. A group of structural engineers collaborated with us to develop two tree models based on data from over 1900 seismic rehabilitation projects.

The first model is designed for use early in developing budget estimates when specific building details are not yet available. This model requires information about a building's original year of construction, its size, its structural system, the seismic zone in which it resides, and the rehabilitation performance objective. We summarize the 11 relevant predictor variables in Table 30.2.

The regression tree is presented in Fig. 30.2. We see that the first split is based on the building's original date of construction. As one might expect, rehabilitation tends to be more costly for older buildings. Regardless of the age of the building, the cost estimate is refined based on the purpose of the rehabilitation effort. Far more expense is required to prepare a building for immediate occupancy than for other purposes. Further down the tree, these cost estimates may be adjusted based on the building's structural characteristics, size, and location.

The second model is used to refine these estimates as more comprehensive data become available. In addition to the basic information included in the smaller model, this larger model uses information about occupancy, number of floors, diaphragm type, the rehabilitation project scope, and other details. More detailed information about the data set used to build these models can be found in FEMA [30.3,4].

30.1.3 Outline

In the rest of this chapter, we will review various tree-based methods for classification and prediction. Section 30.2 details the classification and regression trees (CART) method [30.5] and discusses issues common to all tree-building algorithms. Section 30.3 outlines competing methods, including QUEST [30.6], CRUISE [30.7], GUIDE [30.8], C4.5 [30.9], and chi-square automatic interaction detection (CHAID) [30.10]. Section 30.4 introduces ensemble methods. Finally, Sect. 30.5 discusses briefly how tree methods compare to a broader spectrum of classification and prediction methods.

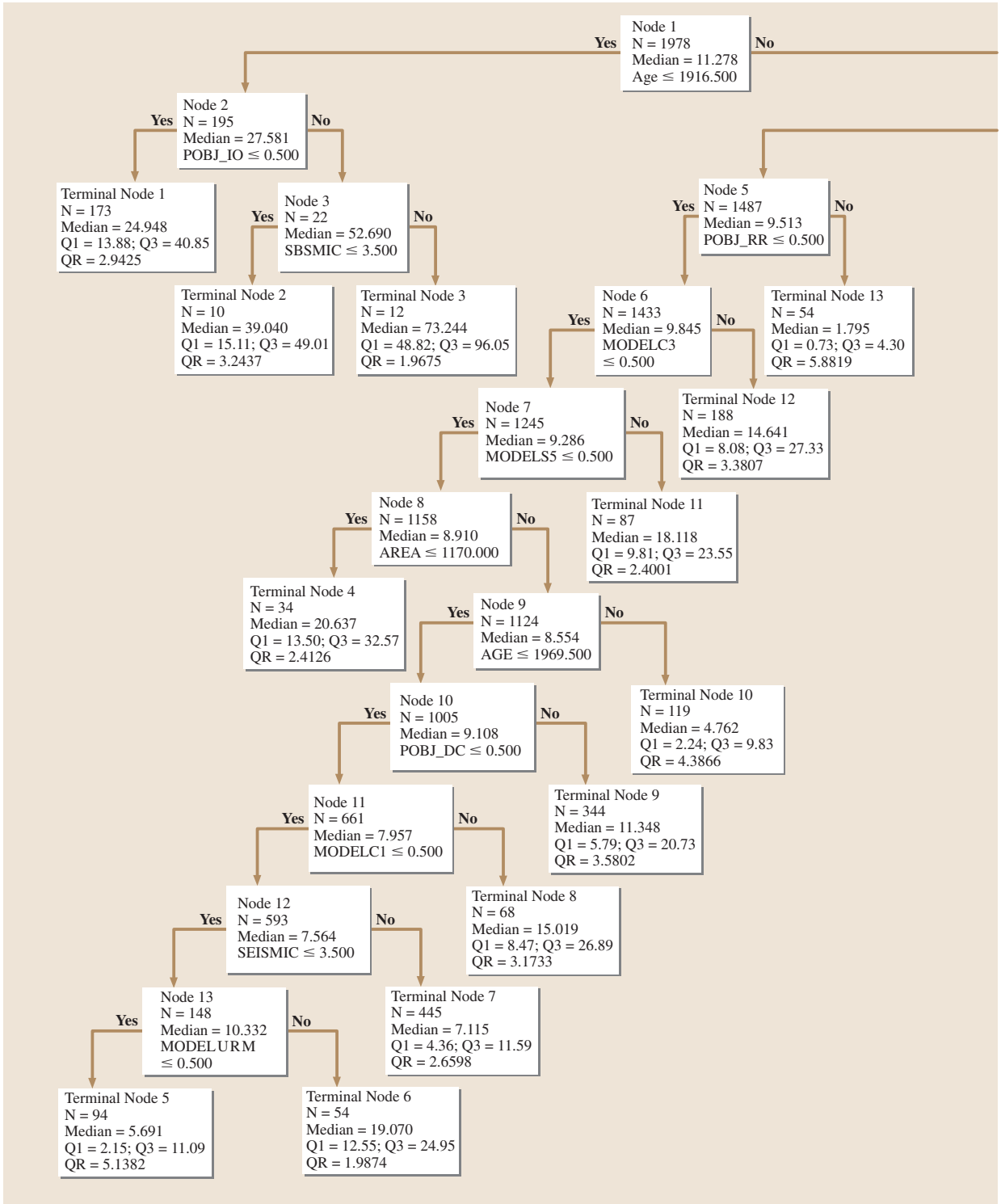
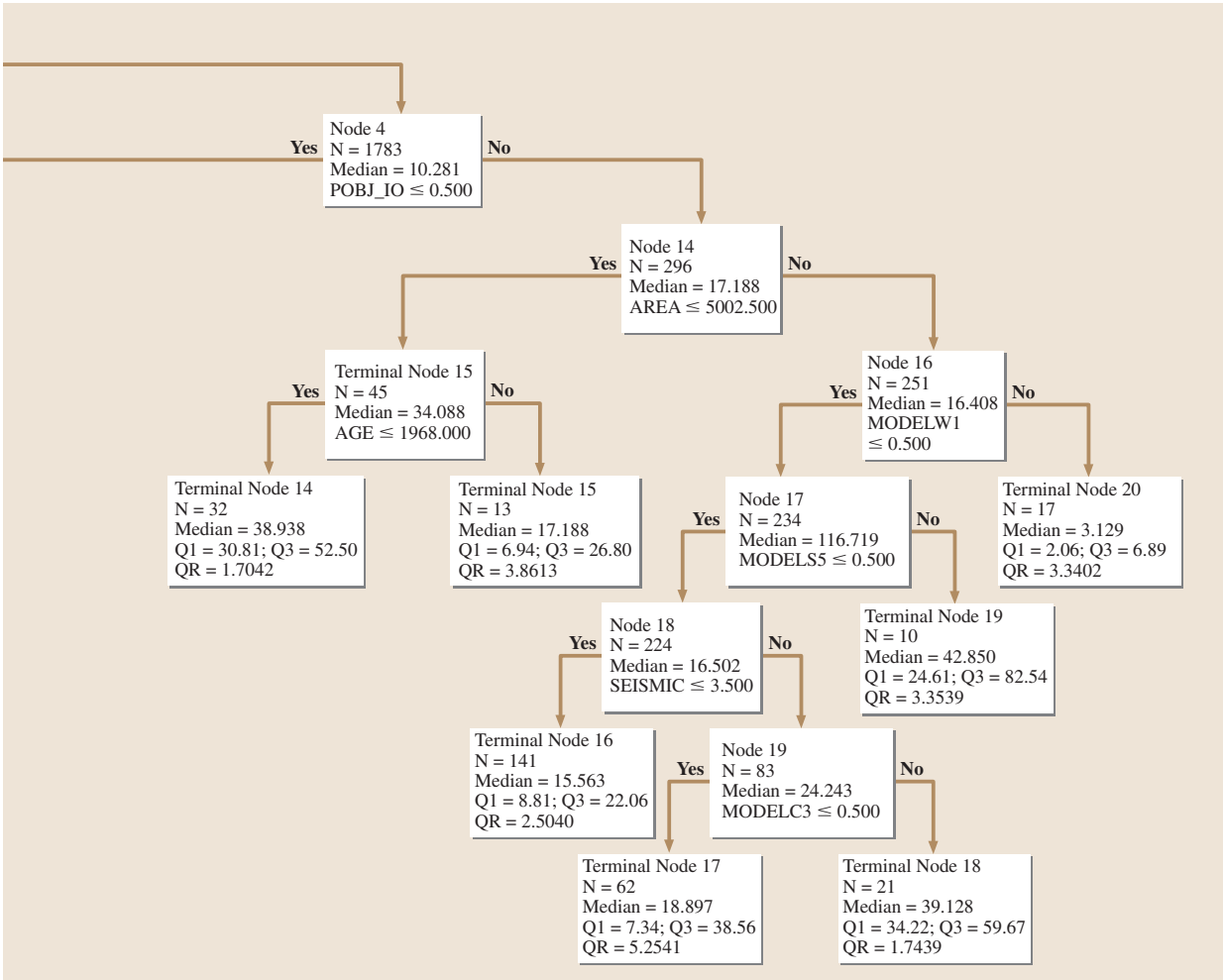


Fig. 30.2 FEMA seismic rehabilitation cost estimator



30.2 Classification and Regression Tree (CART)

30.2.1 Introduction

A widely used tree-based method and software is called CART, which stands for classification and regression tree [30.5]. CART is based on statistical methodology developed for classification with categorial outcomes or regression with continuous outcomes. We shall start with classification trees in Sect. 30.2.2 and 30.2.3 and then discuss the regression tree in Sect. 30.2.4.

Take the iris data classification problem [30.11] as an example. The iris data set contains the lengths and widths of sepals and petals of three types of irises: Setosa, Versicolor, and Virginica. The purpose of the analysis is to learn how one can discriminate among the three types of flowers, Y, based on four measures of width and length of petals and sepals, denoted by X_1, X_2, X_3, and X_4, respectively. Figure 30.3 presents the classification tree constructed by CART. The whole sample sits at the top of the tree. The tree first splits the entire sample into two groups at $X_2 = 2.45$. Observations satisfying the condition $X_2 < 2.45$ are assigned to the left branch and classified as Setosa, while the other observations ($X_2 \geq 2.45$) are assigned to the right branch and split further into two groups at $X_1 = 1.75$.

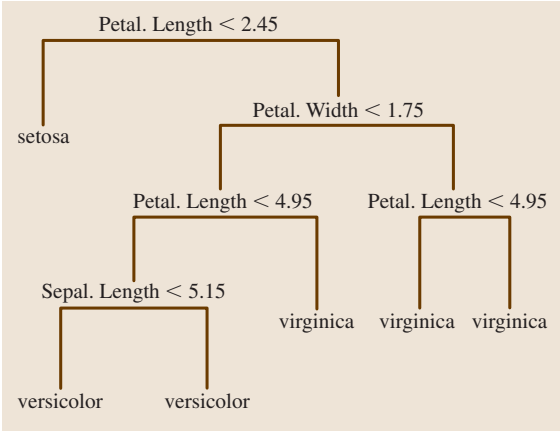


Fig. 30.3 A classification tree for the iris data

At the end, the tree partitions the whole sample into six exclusive subgroups (terminal nodes in the tree). This tree indicates that a good classification rule can be constructed based on the width and length of the petal, and the length of the sepal. The binary tree structure also makes the classification rule easily understood. For example, if the sepal length of an iris with unknown type is 3 cm, its petal length is 4 cm and width is 1.3 cm, then this iris will be classified as a Versicolor iris.

The basic idea of CART is to first grow a very large and complicated tree classifier that explains the sample very accurately but may have poor generalization characteristics, and then prune this tree using cost-complexity pruning to avoid overfitting but still with good accuracy. The CART algorithm grows the classification tree by recursive binary partitioning the sample into subsets. It first splits the entire sample into two subsets, and classifies the observations in each subset using the majority rule. Other class assignment rules can be derived based on preassigned classification costs for different classes. Then one or both of these subsets are split further into more subsets, and this process is continued until no further splits are possible or some stopping rule is triggered. A convenient way to represent this recursive binary partition of the feature space is to use a binary tree like the one in Fig. 30.3, in which subsets are represented by nodes.

30.2.2 Growing the Tree

Let us first look at how CART grows the tree, i. e., how to determine the splitting variable and the split point at each partition. The fundamental idea is to select each split of a node so that the observations in each of the descendant nodes are *purer* than those in the parent node.

Consider a classification problem with a categorical response Y taking values $1, 2, \ldots, K$, and p predictors $X_1, \ldots, X_p$ based on a sample of size N. At node m, which contains a subset of N_m observations, define the node proportion of class k by

$$\hat{p}_m(k) = \frac{1}{N_m} \sum_{i=1}^{N_m} I(y_i = k), \quad k = 1, \ldots, K ,$$

where $I(A) = 1$ when condition A is satisfied and 0 otherwise.

Before discussing how CART splits at a node, we first describe how it classifies a node. In its basic form, CART classifies observations in node m to the majority class $k(m) = \arg\max_k \left[\hat{p}_m(k)\right]$. A more general rule is to assign node m to class $k(m) = \arg\min_k [r_k(m)]$, where $r_k(m)$ is the expected misclassification cost for class k. Letting $\pi_m(k)$ be the prior probability of node m as class k, and $c(i|j)$ be the cost of classifying a class j case as a class i case that satisfies $c(i|j) \geq 0$ if $i \neq j$ and $c(i|j) = 0$ if $i = j$, we have

$$r_k(m) = \sum_j c(k|j)\pi_m(j) .$$

The application of this rule takes into account the severity of misclassifying cases to certain class. If the misclassification cost is constant and the priors $\pi_m(j)$ are estimated by the node proportions, it converts back to the basic form.

CART has two types of splitting criteria: the Gini criterion and the twoing criterion. In general, for a nominal outcome variable, either criterion can be used; for an ordinal outcome variable, the twoing criterion is used.

Gini Criterion

By the Gini criterion, we seek the splitting variable and the split point for node m by maximizing the decrease in the Gini index. The Gini index is an impurity measure defined as a nonnegative function of node proportions $\hat{p}_m(k), k = 1, \ldots, K$,

$$i(m) = \sum_{k=1}^{K} \hat{p}_m(k)\left[1 - \hat{p}_m(k)\right] = 1 - \sum_{k=1}^{K} \left[\hat{p}_m(k)\right]^2 . \tag{30.1}$$

This impurity measure attains its minimum when all cases at a node belong to only one class, so $i(m) = 0$ defines node m as a pure node. Let m_L and m_R be the left and right branches resulting from splitting node m

on predictor x_j, and q_L and q_R be the proportion of cases in node m classified into m_L and m_R, respectively. For each predictor x_j, the algorithm finds the split by maximizing the decrease in the impurity measure

$$\Delta i_j(t,m) = i(m) - [q_L i(m_L) + q_R i(m_R)] \,. \tag{30.2}$$

This is equivalent to minimizing the weighted average of the two child nodes' impurity measures, $q_L i(m_L) + q_R i(m_R)$. When x_j is continuous or ordinal, m_L and m_R are given by $x_j < t$ and $x_j \geq t$ for a splitting point t, and the solution of t can be obtained quickly; if x_j is nominal with a large number of levels, finding the split point t by exhaustive subset search can be computationally prohibitive. The computer program CART only searches over all possible subsets of a categorical predictor for a limited number of levels. The CART algorithm proceeds with a greedy approach that scans through all predictor variables to find the best pair (j, t) with the largest decrease in $\Delta i_j(t, m)$.

Possible choices of $i(m)$ include

- Cross-entropy or deviance:

$$\sum_{k=1}^{K} \hat{p}_m(k) \log \hat{p}_m(k) \,. \tag{30.3}$$

- Misclassification error:

$$\frac{1}{N_m} \sum_{i=1}^{N_m} I\left[y_i \neq k(m)\right] \,. \tag{30.4}$$

The cross-entropy measure (30.3) was used in the early development of CART but the Gini index was adopted in later work. The misclassification error measure (30.4) is typically used during the pruning stage (Sect. 30.2.3). For further discussion of the impurity measures, we refer to *Hastie* et al. [30.12].

Twoing Rule

Under the second splitting criterion, the split at a node m is determined by minimizing the twoing rule

$$q_L q_R \left[\sum_{k=1}^{K} |\hat{p}_{m_L}(k) - \hat{p}_{m_R}(k)|\right]^2 \,.$$

When K is large, twoing is a more desirable splitting criterion.

Comparisons between the Gini and twoing splitting criteria have shown only slight differences, but the Gini criterion is preferred by the inventors of CART and implemented as the default option in the commercial CART software by Salford Systems.

A tree continues to grow until either (1) there is only one observation in each of the terminal nodes, or (2) all observations within each terminal node have an identical distribution of independent variables or dependent variable, making splitting impossible, or (3) it reaches an external limit on the number of observations in each terminal node set by the user.

30.2.3 Pruning the Tree

Growing a very large tree can result in overfitting, that is, the tree classifier has small classification errors on the training sample, but may perform poorly on a new test data set. To avoid overfitting but still capture the important structures of the data, CART reduces the tree to an optimal size by *cost-complexity pruning*. Suppose the tree-growing algorithm stops at a large tree $T_{\max}$. The size of $T_{\max}$ is not critical as long as it is large enough. Define a subtree $T \subset T_{\max}$ to be any tree that can be obtained by pruning $T_{\max}$, that is, collapsing any number of its nodes. The idea is first to find subtrees $T_\alpha \subset T_{\max}$ for a given tuning parameter $\alpha \geq 0$ that minimize the cost-complexity criterion

$$R_\alpha(T) = R(T) + \alpha|T| = \sum_{m=1}^{|T|} N_m i(m) + \alpha|T| \,, \tag{30.5}$$

where m indexes the terminal nodes, $|T|$ is the number of terminal nodes in tree T, and N_m and $i(m)$ are the number of observations and the impurity measure of node m, respectively. Then the optimal tree is selected from this sequence of T_αs. The cost-complexity criterion is a combination of the misclassification cost of the tree, $R(T)$, and its complexity $|T|$. The constant α can be interpreted as the complexity cost per terminal node. If α is small, the penalty for having a larger tree is small and hence T_α is large. As α increases, $|T_\alpha|$ also increases. Typically, the misclassification error impurity measure (30.4) is used in pruning the tree. Equation (30.5) presents a special form of the misclassification cost $R(T)$ when the cost of misclassifying an observation of class j to class i is the same for all $i \neq j$. Other misclassification cost functions $R(T)$ can be applied; see *Breiman* et al. [30.5], but our description of the algorithm will be based on (30.5).

CART uses *weakest-link pruning* to find the T_αs. The algorithm successively collapses the branch that produces the smallest per-node increase in $R(T)$ from the bottom up and continues until it produces the single-node (root) tree. This gives us a sequence of nested subtrees $\{T_0, T_1, T_2, \ldots, T_I\}$ with decreasing

complexity and increasing cost. It is shown in *Breiman* et al. [30.5] that this sequence of subtrees is characterized by distinct and increasing α_is and the α corresponding to the optimal size tree can be found from $\{\alpha_i | i = 0, \ldots, I\}$.

The weakest-link pruning works as follows. Define T_m as a branch of T_{i+1} containing a node m and its descendants. When T_i is pruned at node m, its misclassification cost increases by $R(m) - R(T_m)$, while its complexity decreases by $|T_m| - 1$. Hence the ratio

$$g_i(m) = \frac{R(m) - R(T_m)}{|T_m| - 1}$$

measures the increase in misclassification cost per pruned terminal node, and T_{i+1} is obtained by pruning all nodes in T_i with the lowest value of $g_i(m)$, i. e., the weakest link. The α associated with tree T_i is given by $\alpha_i = \min_m g_i(m)$ and it is easily seen that $\alpha_i < \alpha_{i+1}$. The first tree T_0 is obtained by pruning $T_{\max}$ of those branches whose $g_0(m)$ value is 0. Starting with T_0, the cost-complexity pruning algorithm initially tends to prune off large branches with many terminal nodes. As the trees get smaller, it tends to cut off fewer at a time. The pruning stops when the last subtree T_I is the root tree. These recursive pruning steps are computationally rapid and require only a small fraction of the total tree construction time.

CART then identifies from $\{T_i | i = 0, 1, \ldots, I\}$ the optimal subtree as the one with the minimal classification error (0-SE rule) or the smallest tree within one standard error of the minimum error rate (1-SE rule). The classification error of each subtree T_i can be estimated using test samples when data are sufficient or V-fold cross-validation. The reason for using the 1-SE rule is to favor smaller trees with estimated misclassification errors close to that of the minimum error tree. The 1-SE rule is good for small data sets, whereas the 0-SE rule works better on large data sets. With sufficient data, one can simply divide the sample into learning and test sub-samples. The learning sample is used to grow $T_{\max}$ and to obtain the subsequence $\{T_i | i = 0, 1, \ldots, I\}$. The test sample is then used to estimate the misclassification error rate for the T_is.

When the data are insufficient to allow a good-sized test sample, CART employs cross-validation to estimate the misclassification rate. Cross-validation is a computationally intensive method for validating a procedure for model building, which avoids the requirement for a new or independent validation data set. For V-fold cross-validation, CART proceeds by dividing the learning sample into V parts, stratified by the dependent variable, to assure that a similar outcome distribution is present in each of the V subsets of data. CART takes the first $V - 1$ parts of the data, constructs the auxiliary trees for $\{T_i | i = 0, 1, \ldots, I\}$ characterized by the α_is, and uses the remaining data to obtain initial estimates of the classification error of selected subtrees. The same process is then repeated on other $V - 1$ parts of the data. The process repeats V times until each part of the data has been held in reserve one time as a test sample. The estimates of the classification errors for $\{T_i | i = 0, 1, \ldots, I\}$ are then given by averaging their initial estimates over V artificial test samples.

Many other pruning methods are also available for decision trees, such as reduced error pruning (REP), pessimistic error pruning (PEP), minimum error pruning (MEP), critical value pruning (CVP) and error-based pruning (EBP). *Esposito* et al. [30.13] provides a comprehensive empirical comparison of these methods.

30.2.4 Regression Tree

CART constructs a regression tree when the outcome variable is continuous. The process of constructing a regression tree is similar to that for a classification tree, but differs in the criteria for splitting and pruning. CART constructs the regression tree by detecting the heterogeneity that exists in the data set and then purifying the data set. At each node, the predicted value of the dependent variable is a constant, usually as the average value of the dependent variable within the node. An example of a regression tree is given in Fig. 30.4. The analysis tried to construct a predictive model of cen-

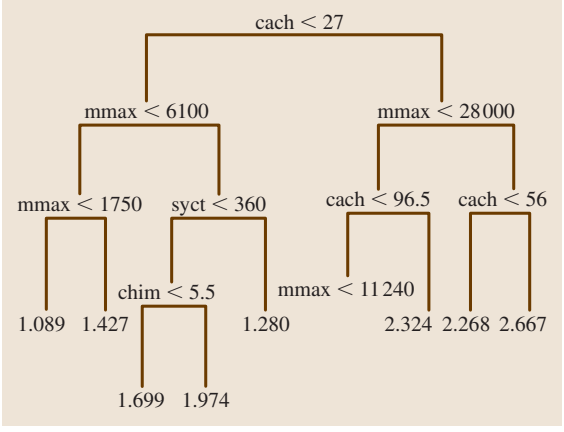


Fig. 30.4 A regression tree for predicting CPU performance

tral processing unit (CPU) performance using nine CPU characteristics (Table 30.3) based on a learning sample of 209 CPUs [30.14].

When CART grows a regression tree, it determines the splitting variable and split point by minimizing the mean square error (MSE) or the mean absolute deviation from the median. Since the mechanisms for the two rules are similar, we only describe the former. Under this circumstance, the node impurity is measured by

$$i(m) = \frac{1}{N_m} \sum_i \left[y_i - \bar{y}(m) \right]^2 , \tag{30.6}$$

where $\bar{y}(m)$ is the average value of the dependent variable at node m. The best split (j, t) is hence determined by solving

$$\min_{j,t} \left\{ \sum_{i \in m_L} \left[y_i - \bar{y}(m_L) \right]^2 + \sum_{i \in m_R} \left[y_i - \bar{y}(m_R) \right]^2 \right\} , \tag{30.7}$$

where m_L is the left descendent node given by $x_j < t$, and m_R is for the right branch. An alternative criterion to (30.7) is the weighted variance

$$p_L i(m_L) + p_R i(m_R) ,$$

where p_L and p_R are the proportions of cases in node m that go left and right, respectively. Correspondingly, the cost-complexity criterion in the pruning process also adopts (30.6).

30.2.5 Some Algorithmic Issues

In this section, we discuss several algorithmic issues of CART that are important in practice.

Estimating Within-Node Classification Error

In practice, the users desire to know not only the class assignment from a CART tree for any future case, but also about a classification error associated with this prediction. This classification error can be represented by the probability of misclassification given that the case falls into a particular terminal node. We denote this value by $r(m)$ if a case falls into terminal node m. A naive estimate of $r(m)$ is the proportion of cases that are misclassified by the tree constructed from the entire sample, as shown in (30.4). This estimate however can be misleading since it is computed from the same data used in constructing the tree. It is also unreliable if the terminal node m is tracked down through many splits from the root and has a relatively small number of observations.

Table 30.3 Characteristics of CPUs

Variable	Definition
name	Manufacturer and model
syct	Cycle time in nanoseconds
mmin	Minimum main memory in kilobytes
mmax	Maximum main memory in kilobytes
cach	Cache size in kilobytes
chmin	Minimum number of channels
chmax	Maximum number of channels
perf	Published performance on a benchmark mix
	relative to an IBM 370/158-3
estperf	Estimated performance by the authors

Breiman et al. [30.5] proposed an ad hoc estimate that is significantly better than the naive one,

$$\hat{r}(m) = \hat{r}_o(m) + \frac{\epsilon}{N_m + \lambda} , \tag{30.8}$$

where $\hat{r}_o(m)$ is defined in (30.4), N_m is the size of node m, and ϵ and λ are constants to be determined below. Define the resubstitution classification error of the tree T by aggregating the classification errors across all terminal nodes as follows,

$$\hat{R}_o(T) = \sum_{m=1}^{|T|} \hat{r}_o(m) N_m .$$

Denote the cross-validated classification error of tree T by $\hat{R}_{CV}(T)$. Then the constants λ and ϵ are obtained from the following equations

$$\lambda \sum_{m=1}^{|T|} \frac{N_m}{N_m + \lambda} = \frac{\hat{R}_{CV}(T) - \hat{R}_o(T)}{2 \min_V \hat{R}_{CV}(T_V)} ,$$

$$\epsilon = 2\lambda \min_V \hat{R}_{CV}(T_V) ,$$

where $\min_V \hat{R}_{CV}(T_V)$ is the minimum obtained during V-fold cross-validation. If $\hat{R}_{CV}(T) \le \hat{R}_o(T)$, the naive estimate (30.4) is used.

Splitting on a Linear Combination of Variables

Sometimes, the data are intrinsically classified by some hyperplanes. This can possibly challenge the tree-based method in its original form using binary partitions, which tends to produce a large tree in trying to approximate the hyperplanes by many hyperrectangular regions. It is also very hard for the analysts to recognize the neat linear structure from the output. The CART algorithm deals with this problem by allowing splits over

linear combination of predictor variables $\sum_j a_j X_j$. The weights a_j and split point t are optimized to minimize the relevant criterion (such as the Gini index). While this can improve the predictive power of the tree, the results are no longer invariant under monotone transformations of individual independent variables. The introduction of linear combinations also causes a loss in interpretability that is viewed as an important advantage of tree-based methods.

Missing Data on Predictors

We often have incomplete data with missing values on some independent variables. We might exclude these incomplete observations from analysis, but this could lead to serious depletion of the learning sample. A common alternative is to impute the missing values [30.15]. CART however uses two different approaches. A simple treatment for categorical predictors is to put the missing values into a new "missing" category. This however puts all observations with missing values into the same branch of the tree, which could be misleading in practice. A more refined approach is to use surrogate splits. This approach makes full use of the data to construct the tree, and results in a tree that can classify cases with missing information. Surrogate variables are constructed as follows. When we consider a split on a predictor x_j with missing values, only the cases containing values of x_j are used, and we find the best split as discussed in Sect. 30.2.2. The first surrogate split is the split on a predictor that most accurately predicts the action of the best split in terms of a predictive measure of association. The second surrogate is the split on another predictor that does second best, and so on; for details see [30.5]. The surrogate splits can cope with missing observations during both the training phase of CART and prediction. If a case has missing values so that the best split is not useable, the next best surrogate split would be used.

Variable Importance

Another nice feature of CART is that it automatically produces a variable ranking. The ranking considers the fact that an important variable might not appear in any split in the final tree when the tree includes another masking variable. If we remove the masking variable, this variable could show up in a prominent split in a new tree that is almost as good as the original. The importance score of a particular variable is the sum of the improvement of impurity measures across all nodes in the tree when it acts as a primary or surrogate splitter.

Instability of Trees

Small changes in the learning sample may cause dramatic changes in the output tree. Thus two similar samples could generate very different classification rules, which is against human intuition and complicates interpretation of the trees. The major reason for this instability is the hierarchical nature of the recursive partitioning. For example, if at some partition, there are surrogate splits that are almost as good as the primary split, the tree could be very sensitive to small changes, because a minor change in the learning sample could cause the surrogate split to become slightly superior to the primary split. This effect in the top nodes can cascade to all their descendant nodes. Aggregating methods, such as bagging [30.16] and boosting [30.17] have been incorporated into the algorithm to mitigate the instability problem, but the improvement comes at the price of sacrificing the simple interpretability of a single tree.

30.2.6 Summary

CART makes no distributional assumptions on any dependent or independent variable, and allows both categorical and continuous variables. The algorithm can effectively deal with large data sets with many independent variables, missing values, outliers and collinearity. Its simple binary tree structure offers excellent interpretability. Besides, CART ranks the independent variables in terms of their importance to prediction power and therefore serves as a powerful exploratory tool for understanding the underlying structure in the data.

However, CART does have limitations. While it takes advantages of the simple binary tree structure, it suffers from instability and has difficulty capturing additive structures. In general, if a parametric statistical model fits the data well and its assumptions appear to be satisfied, it would be preferable to a CART tree.

30.3 Other Single-Tree-Based Methods

30.3.1 Loh's Methods

One drawback of exhaustive-search tree-growth algorithms such as that used in CART is the potential for variable selection bias. In particular, such algorithms tend to choose variables that provide greater numbers of potential splitting points. Hence, continuous variables tend to be favored over categorical variables, and polychotomous variables are selected more frequently than dichotomous ones. These characteristics complicate interpretation of resulting trees, because any insights gained from the tree structure could potentially be clouded by systematic biases toward certain variables. The methods developed by Loh and his coauthors attempt to address this bias issue.

QUEST

Loh and *Shih* [30.6] developed the quick, unbiased and efficient statistical tree (QUEST) algorithm to address this variable selection bias issue. The algorithm is an enhancement of the much earlier fast algorithm for classification trees (FACT) of *Loh* and *Vanichsetaukul* [30.18], which was primarily designed as a computationally efficient alternative to exhaustive-search methods, but still suffered from variable-selection bias in the presence of categorical predictors.

The basic strategy employed by QUEST is to select each splitting variable and its associated split value sequentially rather than simultaneously. To determine the splitting variable at a particular node, a series of statistical tests is performed:

1. Specify an overall level of significance, $\alpha \in (0, 1)$. Let K be the number of variables, and K_1 be the number of continuous and ordinal variables.
2. Identify the variable with the smallest p-value resulting from the appropriate analysis of variance test (for continuous or ordinal variables) or Pearson's χ^2 test (for categorical variables). If this p-value is less than α/K, split on this variable.
3. If the lowest p-value exceeds this threshold, perform Levene's F-test for unequal variances on each continuous/ordinal variable. If the smallest of these p-values from the F-tests is less than $\alpha/(K + K_1)$, split on its associated variable. Otherwise, split on the variable with the smallest p-value in step 2.

The Bonferroni-adjusted thresholds used above is meant to render the potential variable-selection bias negligible.

Once the splitting variable is selected, the split point is needed. If more than two classes are present at the node, they are first combined into two *superclasses* using two-means clustering [30.19]. Then, a modified quadratic discriminant analysis is employed to select the split point. Categorical variables must be transformed into ordered variables before this split can be performed. This is accomplished by recoding the represented categories as 0–1 dummy vectors and projecting them onto their largest discriminate coordinate.

The algorithm described above focuses on univariate splits. However, as with CART, QUEST can also be used to build trees with linear-combination splits. Generally, QUEST trees based on linear-combination splits tend to be shorter and more accurate than those based on univariate splits.

The QUEST package may be obtained from http://www.stat.wisc.edu/~loh/loh.html. The full package includes an exhaustive-search algorithm to mimic basic CART, and offers options for pruning or stopping rules.

CRUISE

Kim and *Loh* [30.7] extended the unbiased variable selection idea beyond the capabilities of QUEST. First, while QUEST forces binary splits at each node, classification rule with unbiased interaction selection and estimation (CRUISE) allows multiway splitting. Moreover, CRUISE includes look-ahead methods for detecting two-variable interactions during variable selection.

Multiway splitting offers two key advantages over binary splitting. First, although any multiway split can be represented by a series of binary splits, trees that allow multiway splits are often shorter and thus more easily interpreted. Second, Kim and Loh demonstrate that, with binary trees, some dependent variable categories can be completely dropped after pruning. For example, a tree intended to classify cases into two categories might ultimately include paths to only two of the classes of interest. Trees employing multiway splits are less apt to *losing* categories in this manner.

Another key benefit of CRUISE is the inclusion of look-ahead methods for detecting two-variable interactions during variable selection. CRUISE contains two methods for splitting-variable selection: 1D and 2D. The

1D method is similar to what is used in the QUEST algorithm. p-values are obtained from F-tests for continuous and ordinal variables and from Pearson's χ^2 tests for categorical variables. If the smallest p-value is significant, its associated variable is selected for the split. Otherwise, Levene's test for unequal variances is carried out for the continuous and ordinal variables to select the splitting variable. A major drawback of the 1D method is that, because analysis of variation (ANOVA) and Levene's tests do not look ahead, strong interactions are often completely overlooked. In addition, because these tests restrict their attention to differences in means and variances, other distributional differences may remain unnoticed.

The 2D method uses contingency tables to remedy these problems. First, consider interaction detection. Given a pair of categorical variables, category pairs are tabulated against classifications. Then, Pearson's χ^2 test for independence is performed. If a strong interaction is present between the two categorical variables, the test is likely to result in a low p-value. Interactions involving continuous variables are detected similarly. Prior to testing, each continuous variable is transformed into a dichotomous variable by partitioning its domain at the median.

The same idea is applied to identify marginal distributional effects. For each categorical variable, Pearson's χ^2 test for independence is performed. Continuous variables are handled similarly, first transforming them into four-category variables by partitioning at their quartiles. The basic idea is that the one- or two-variable table with the smallest p-value should determine the splitting variable. However, this simple procedure would be somewhat biased toward categorical variables, so Kim and Loh employ a bootstrap adjustment prior to variable selection.

Once the splitting variable has been selected, CRUISE uses linear discriminant analysis (LDA) to determine the splitting points. Since LDA is best applied to normally distributed data, Kim and Loh apply a Box–Cox transformation to the selected variable prior to running the discriminant analysis. Categorical variables must be converted to their discriminant coordinate values before this process is carried out. A shift transformation may be needed to produce the positive-valued inputs required for the Box–Cox procedure. Split points are converted back to the original scale when constructing the tree.

Our description thus far assumes the availability of complete data, but an important advantage of CRUISE is the elimination of the variable-selection bias that often results from the treatment of missing data. Kim and Loh note that, because CART uses proportions rather than sample sizes to determine variable selections, the procedure is biased toward variables with more missing data. CRUISE on the other hand, through its use of statistical tests that take account of sample size, does not encounter this type of bias. This bias may not be critical if it does not affect the overall predictive quality of the tree, but it may have a large impact on the interpretation of CART's variable-importance measures.

GUIDE

With generalized, unbiased interaction detection and estimation (GUIDE), *Loh* [30.8] expanded unbiased variable selection to regression tree applications. GUIDE includes procedures for weighted least squares, Poisson regression and quantile regression. In addition, categorical variables may be used for prediction through dummy coding, or they may be restricted to node-splitting.

30.3.2 Quinlan's C4.5

Quinlan [30.9, 20] wrote his first decision tree program in 1978 while visiting Stanford University. His iterative dichotomizer 3rd (ID3) and its replacement, C4.5, programs have served as the primary decision tree programs in the artificial-intelligence and machine-learning communities. He attributes the original ideas to the concept learning systems of *Hunt* et al. [30.21].

Splitting Rules

Suppose a node T contains $|T|$ observations that fall into K classes. Letting $p_k(T)$ represent the proportion of these cases belonging to class k, we define the *information* contained within T (also known as the *entropy* of T) by:

$$\text{Info}(T) = -\sum_{k=1}^{K} p_k(T) \times log_2 \left[p_k(T) \right] .$$

Now suppose that a candidate variable X partitions T into n smaller nodes, $T_1, T_2, \ldots, T_n$. The information of T given the value of X is given by the weighted average of the information contained within each subnode:

$$\text{Info}(X, T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \text{Info}(T_i) .$$

Therefore, the *information gain* provided by the split is simply

$$\text{Gain}(X, T) = \text{Info}(T) - \text{Info}(X, T) . \qquad (30.9)$$

ID3 selects attributes and splits to maximize the information gain at each node. However, this procedure tends to heavily favor variables with many categories. To compensate for this effect at least partially, C4.5 instead uses the *gain ratio* criterion. The gain ratio of a split is defined as the ratio of the information gain to the information contained in the resulting split:

$$\text{GainRatio}(X, T) = \frac{\text{Gain}(X, T)}{\text{SplitInfo}(X, T)}\,, \qquad (30.10)$$

where

$$\text{SplitInfo}(X, T) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2\left(\frac{|T_i|}{|T|}\right).$$

The C4.5 algorithm creates binary splits on continuous variables and multiway splits on categorical variables. To determine the best splits on categorical variables, each category is first assigned to a unique branch. Then, pairs of branches are iteratively merged until only two branches exist. The split with the maximum gain ratio among those observed becomes the candidate split for that variable. This search method is, of course, heuristic and might not actually find the categorical split with the largest gain ratio. On the other hand, searches on continuous variables always find the best possible binary split. To determine the ultimate splitting variable, the algorithm first restricts its choices to those variables achieving at least average information gain (30.9). The split is then selected to maximize the gain ratio (30.10) among these choices.

Variable-Selection Bias

Even if the gain ratio is used as an alternative to straight information gain to alleviate the algorithm's bias toward continuous variables, this original remedy is far from perfect. *Dougherty* et al. [30.22] demonstrated that, for many data sets, the predictive performance of C4.5 was improved by first discretizing all continuous variables. This result suggested that the existing selection method was biased toward continuous variables.

In C4.5 release 8, *Quinlan* [30.23] introduces a complexity-cost parameter into the information gain expression for continuous variables. For a continuous variable with n distinct values, the information gain is redefined as

$$\text{Gain}(X, T) = \text{Info}(T) - \text{Info}(X, T) - \log_2(n-1)/|T|\,.$$

Effectively, each continuous variable is penalized for the information required to search among its numerous potential splitting points.

Missing Values

The description of C4.5 has thus far assumed complete data. Cases with missing values for a particular variable are excluded from the split search on that variable, and also from the numerator of the gain ratio. The entropy of the split is computed as if missing values constitute an additional branch.

When a missing value prevents the application of a splitting rule to a new case, the case is replaced by weighted replicates, each being assigned to a different branch. The weights are equal to the proportion of non-missing training cases assigned to that branch. Class probabilities for the original case are based on the weighted sum of the probabilities of the generated observations.

Pruning

Quinlan [30.9] advocates retrospective pruning instead of stopping rules. If enough data were available, the pruning process would use data withheld from training the tree to compare error rates of candidate sub-trees. The software does not assume that data are withheld from training, so it implements *pessimistic* pruning. In each node, an upper confidence limit of the number of misclassified cases is estimated assuming a binomial distribution around the observed number of misclassified cases. The confidence limit serves as an estimate of the error rate on future data. The pruned tree minimizes the sum over leaves of upper confidence limits.

Decision Rules

C4.5 includes the capability to convert its decision trees to an equivalent simplified set of decision rules. Decision rules are often preferred to the tree structure because their interpretation is very straightforward. Given a decision tree, rule-set generation proceeds as follows:

1. Convert every decision tree path to a decision rule. (Each decision encountered along a path becomes a *test* in the resulting decision rule.)
2. Prune each decision rule by removing as many tests as possible without reducing its accuracy.
3. Track the estimated accuracy of each resulting rule, and classify new items based on high-accuracy rules first.

Improvements in C5.0/See5

C5.0 and See5 are the current commercial implementations of Quinlan's methods. These programs offer several enhancements to C4.5, including the ability to specify unequal misclassification costs, the application of fuzzy splits on continuous variables, and boosting trees.

30.3.3 CHAID

Chi-square automatic interaction detection (CHAID) is a parametric recursive partitioning technique that builds non-binary classification trees. It was originally developed by *Kass* [30.10] to handle categorical predictors only. Continuous predictors need to be discretized into a number of categories with approximately equal number of observations. In dealing with missing values on predictors, CHAID simply places them in an additional category. The algorithm employs a sequential merge-and-split procedure based on significance tests on predictor variables to generate node splits and determine the size of a tree. It is worth noting that CHAID differs from CART in that it determines where to stop in tree growth rather than using retrospective pruning after growing an oversized tree.

CHAID produces non-binary trees that are sometimes more succinct representations than equivalent binary trees. For example, it may yield a split on an income variable that divides people into four income groups according to some important consumer-behavior-related variable (e.g., types of cars most likely to be purchased). In this case, a binary tree is not an efficient representation and can be hard to interpret. On the other hand, CHAID is primarily a step-forward model-fitting method. Known problems with forward stepwise regression fitting models are probably applicable for this type of analysis.

30.3.4 Comparisons of Single-Tree-Based Methods

We have discussed six single-tree methods, viz. CART, C4.5, CHAID, CRUISE, GUIDE and QUEST. Table 30.4 lists the features offered by these six methods. Among these methods, GUIDE is a regression tree method, CHAID, CRUISE and QUEST are classification tree methods, and CART and C4.5 deal with both classification and regression problems.

Empirical comparisons on real data sets [30.24] showed that, among all these methods, there is none

Table 30.4 Comparison of tree-based algorithms

Feature	CART	C4.5	CHAID	CRUISE	QUEST	GUIDE
Dependent variable						
Discrete	x	x	x	x	x	
Continuous	x	x				x
Split at each node						
Binary	x				x	x
Multiple		x	x	x		
Split on linear combinations	x			x	x	x
Searching splitting variable						
Exhaustive	x		x	x	x	x
Heuristic		x	x			
Splitting criterion						
Impurity measure	x	x				
Twoing rule	x					
Statistical test			x	x	x	x
Split variable selection						
Unbiased selection				x	x	x
Pairwise interaction detection				x		
Tree size control						
Cost-complexity pruning	x			x	x	x
Pessimistic error pruning		x				
Stopping rules			x			x
Missing data						
Surrogate	x			x		
Imputation				x	x	x
An additional level		x	x			

Table 30.5 Data-mining software for tree-based methods

Software	CART	C4.5	CHAID	Software Provider
AnswerTree	x	x	x	SPSS Inc.
Clementine		x		Integral Solutions, Ltd.
Darwin	x			Thinking Machines, Corp.
Enterprise Miner	x	x	x	SAS Institute
Gain Smarts	x		x	Urban Science
MineSet	x		x	Silicon Graphics, Inc.
Model 1	x		x	Group 1/Unica Technologies
Model Quest	x			AbTech Corp.
CART	x			Salford Systems
R	x			R Foundation for Statistical Computing
S-Plus	x			MathSoft
See5		x		RuleQuest Research

that is absolutely superior to the others in terms of accuracy, complexity, interpretability and computation time. There is no significant difference in terms of prediction accuracy among these methods. Therefore, users may choose algorithms based on desired features for their applications, e.g., binary-split, multi-split, split on combination of variables. C4.5 tends to produce trees with many more leaves than other algorithms possibly due to under-pruning. In general, the multi-split tree methods (C4.5, CHAID, CRUISE) take more computation time than the binary-split methods. In problems with mixtures of continuous variables and categorical variables having different numbers of levels, methods such as QUEST, CRUISE, and GUIDE may be preferable because they are likely to protect against variable-selection bias.

CART, CHAID and C4.5 have been implemented in several commercial software platforms; see the list of software providers in Table 30.5. Free software for CRUISE, GUIDE, and QUEST can be obtained from the website http://www.stat.wisc.edu/~loh/. An earlier version of C4.5 is available free of charge http://www.cse.unsw.edu.au/~quinlan/.

30.4 Ensemble Trees

Instability of single trees provides room for improvement by ensemble methods. Ensemble methods create a collection of prediction/classification models by applying the same algorithm on different samples generated from the original training sample, then make final predictions by aggregating (voting) over the ensembles. It has been shown to improve the prediction/classification accuracy of a single model with significant effectiveness; see *Bauer* and *Kohavi* [30.25], *Breiman* [30.16, 26, 27], *Dietterich* [30.28], and *Freund* and *Schapire* [30.29]. The mechanism used by the ensemble methods to reduce prediction errors for unstable prediction models, such as trees, is well understood in terms of variance reduction due to averaging [30.12]. In this section, we will discuss two ensemble tree methods: boosting decision trees [30.17] and random forests [30.26, 27] that are motivated by boosting [30.29] and bagging [30.16], the two most widely used ensemble techniques today. However, it should be realized that better performance of ensemble trees comes at the price of sacrificing the explicit structure of a single tree and hence becoming less interpretable.

30.4.1 Boosting Decision Trees

Boosting was originally developed to improve the performance of binary classifiers. In his original boosting algorithm, *Schapire* [30.30] enhances a *weak learner* (i. e., a binary classifier with slightly better performance than random guessing) by using it to train two additional classifiers on specially filtered versions of the training data. The first new classifier is trained on cases for which the original weak learner performs no better than random guessing. The second new classifier is trained on cases where the first two learners disagree. In this way, each successive learner is trained on cases which are increasingly difficult to classify. The final boosted classifier is obtained by taking the majority vote of the orig-

inal weak learner and its two subsequent derivatives. Schapire's *strength of weak learnability* theorem proves that this simple boosted classifier always improves on the performance of the original weak learner.

In later work, *Freund* [30.31] improved on the performance of Schapire's method by expanding to a much larger ensemble of combined weak learners and again employing the majority vote principle. Subsequent theoretical improvements led to the more flexible AdaBoost algorithm [30.29] and various derivatives.

Our presentation of boosting algorithms and their application to classification and regression trees is based on the example of *Hastie* et al. [30.12].

AdaBoost

We begin by presenting the most popular of the AdaBoost algorithms, AdaBoost.M1 [30.32], which is used for binary classification problems.

Consider a binary classification problem with categories coded as $Y \in \{-1, 1\}$. Given a predictor vector $\boldsymbol{X}$, the classifier $G(\boldsymbol{X})$ takes on values in $\{-1, 1\}$. The error rate on the training sample is given by:

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} I\left[y_i \neq G(x_i)\right]$$

and the expected future prediction error is

$$E_{XY} I\left[Y \neq G(X)\right] \ .$$

The AdaBoost.M1 algorithm proceeds as follows:

1. Initialize the observation weights $w_i = 1/N, i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   a) Fit a classifier $G_m(x)$ to the training data using the weights w_i.
   b) Compute
   $$err_m = \frac{\sum_{i=1}^{N} w_i I\left[y_i \neq G_m(x_i)\right]}{\sum_{i=1}^{N} w_i} \ ;$$
   c) Compute
   $$\alpha_m = \log\left(\frac{1 - err_m}{err_m}\right) \ ;$$
   d) Set $w_i \leftarrow w_i \exp\left\{\alpha_m I\left[y_i \neq G_m(x_i)\right]\right\}$, $i = 1, 2, \ldots, N$.
3. Define the boosted classifier as $G(x)$
   $= \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

This boosting process begins with a weak learner, G_1, which is developed using an unweighted training set. The data are then weighted to deemphasize correctly classified observations and focus on incorrectly classified observations. A new weak classifier, G_2, is then trained from this weighted data. Next, these two classifiers are weighted according to their individual error rates (with the more accurate classifier given greater influence). Based on the weighted performance of the two classifiers, the training data is again reweighted for emphasis on difficult-to-classify observations, and the process iterates. Each new learner, G_m, is thus designed to address increasingly difficult aspects of the classification problem. The final boosted classifier, $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$, is derived from the weighted votes of the M individual weak classifiers. The error rates of the individual weak classifiers G_m tend to increase with each iteration, but prediction from the overall ensemble, G, tends to improve.

Our discussion of boosting thus far applies to classifiers in general. We now narrow our discussion to the particular application of boosting techniques to tree-based models.

Boosting Trees

Boosting procedures in general fit additive expansions of weak classifiers or regressors. In the case of tree models, such expansions have the form

$$f(x) = \sum_{m=1}^{M} \beta_m T(x; \Theta_m) \ ,$$

where the parameter Θ includes information about the structure of each tree.

Such models are fit by minimizing a loss function, L, averaged over the training data, that is, y solving

$$\min_{\{\beta, \Theta\}} \sum_{i=1}^{N} L\left[y_i, \sum_{m=1}^{M} \beta_m T(x_i; \Theta_m)\right] \ .$$

The solution to this problem is approximated using a forward stagewise additive algorithm. The idea is to build the expansion one term at a time. At a given iteration m, the optimal basis tree and scaling coefficient are sought to append to the old expansion f_{m-1}, producing f_m. The algorithm goes as follows:

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:
   a) Compute
   $$(\beta_m, \Theta_m) = \arg\min_{\beta, \Theta} \sum_{i=1}^{N} L\left[y_i, f_{m-1}(x_i) + \beta T(x_i; \Theta)\right] \ .$$

b) Set $f_m(x) = f_{m-1}(x) + \beta_m T(x; \Theta_m)$.

Friedman et al. [30.33] discovered that, under the exponential loss, $L[y, T(x; \Theta)] = \exp[-yf(x)]$, for binary classification problems, the forward stagewise algorithm is equivalent to the AdaBoost.M1 procedure discussed earlier. The expansion $\sum_{m=1}^{M} \alpha_m G_m(x)$ produced by the AdaBoost procedure estimates half the log-odds of $P(Y = 1|x)$. Therefore, taking the sign of this expression provides a reasonable classification rule.

For K-class classification and regression problems, the multiple additive regression trees (MART) procedure is used. MART is based on the gradient tree-boosting algorithm for regression, and can be implemented using a variety of available loss functions; see *Hastie* et al. [30.12] for details.

Selecting Component Tree Sizes

One consideration when applying boosting to tree models is the appropriate size of each weak learner tree. Early tree boosting applications, of which *Drucker* and *Cortes's* [30.17] optical character-recognition problem was the first, applied standard pruning methods to each weak learner in sequence. However, as *Hastie* et al. [30.12] note, this method implicitly prunes each weak learner as if it were the last in the sequence. This can result in poor predictive performance of the ensemble, as well as some unnecessary computations.

A common strategy to avoid this problem is to restrict each tree in the ensemble to a fixed number, J, of terminal nodes. The choice of this parameter is dependent on the problem at hand. Of course, the degree of variable interaction will be affected by the tree size. For example, boosting tree stumps (i. e., trees with only one split) considers no interaction effects, whereas boosting three-node trees can capture two-variable interactions. If each weak learner consists of J terminal nodes, interactions of up to $J-1$ variables may be estimated.

In practice, J is typically determined through trial and error to maximize performance. *Hastie* et al. [30.12] indicate that $4 \le J \le 8$ terminal nodes per tree typically work well, with little sensitivity to choices within that range. For some applications, boosting stumps ($J = 2$) may be sufficient, and very rarely is $J > 10$ needed.

Interpretation

Although boosting trees provides significant improvements in classification and predictive accuracy, these benefits do come at a cost. Because the final model is comprised of the weighted average of many weaker models, we lose the attractive structural interpretability of a single tree.

However, additional useful information can still be gleaned from the data. As we discussed in Sect. 30.2.5, *Breiman* et al. [30.5] provide a measure of the relative importance of predictor variables in a single tree. This measure is easily generalized to the context of boosting. Single-tree importance measures are calculated for each weak learner and averaged over the group. In K-class classification problems, importance measures are generated in this manner for each class. These values can be averaged across classes to obtain overall importance measures for each variable, or across subsets of variables to determine the relevance of each subset in predicting each class.

Once the most relevant variables are identified, certain visualization tools can aid interpretation. *Hastie* et al. [30.12] suggest the use of partial dependence plots to look for interactions between variables.

30.4.2 Random Forest

Breiman [30.26, 27] developed random forests (RF) based on bagging and random feature selection [30.28, 34]. Bagging is a resampling procedure that produces bootstrap samples by randomly sampling with replacement from the original training sample. A random forest is essentially an ensemble of CART trees in which each tree is grown in accordance with a different bootstrap sample. Suppose M bootstrap samples are generated, viewing them as realizations of independent identically distributed (iid) random vectors $\Theta_1, \ldots, \Theta_M$, we denote the random forest by $h(\boldsymbol{x}; \Theta)$ as an ensemble of individual CART trees $h(\boldsymbol{x}; \Theta_j)$, $j = 1, \ldots, M$. For classification problems, the final prediction of the forest is made by majority vote,

$$h(\boldsymbol{x}; \Theta) = \arg\max_k \sum_{j=1}^{M} I\left[h(\boldsymbol{x}; \Theta_j) = k\right] ;$$

for regression problems, the final prediction is obtained by aggregating over M trees, typically using equal weights,

$$h(\boldsymbol{x}; \Theta) = \frac{1}{M} \sum_{j=1}^{M} h(\boldsymbol{x}; \Theta_j) .$$

In accordance with the basic principle of bagging to reduce prediction errors from averaging over the ensemble, better accuracy of the random forest can be obtained by keeping errors of individual trees low, and

minimization of the correlation between multiple trees. Therefore, individual trees are not pruned but grown to maximum depth. Recently, *Segal* [30.35] suggested that this strategy can overfit the data and it is beneficial to regulate tree size by limiting the number of splits and/or the size of nodes for which splitting is allowed. In addition, the correlation of multiple trees can be reduced by random feature selection. Instead of determining the split at a given node in an individual tree using all the predictors, only $m < p$ randomly selected predictors are considered for the split. This also enables the algorithm to build models for high-dimensional data very quickly. Alternatives to this random feature selection include: (1) picking the best out of several random feature subsets by comparing how well the subsets perform on the samples left out of the bootstrap training sample (out-of-bag samples), and (2) using random linear combinations of features in the selected feature subset, i. e., selected features are added together with coefficients that are uniform random numbers on $[-1, 1]$. Due to the large number of simple trees and the minimized correlations among the individual trees, the prediction error of the forest converges toward the error rates comparable to AdaBoost [30.29].

Usually, about one third of the observations are left out of each bootstrap sample. These out-of-bag (oob) observations are used to internally estimate prediction error for future data, the strength of each tree, and correlation between trees; see details in *Breiman* [30.27]. This avoids the cross-validation needed for construction of a single tree and greatly enhances the computational efficiency of random forests.

With random forests, an intuitive measure of variable importance can be computed as follows. In every tree grown in the forest, put down the oob cases and count the number of votes cast for the correct class. Now randomly permute the values of variable m in the oob cases and put these cases down the tree. Subtract the number of votes for the correct class in the variable-m-permuted oob data from the number of votes for the correct class in the untouched oob data. The average of this number over all trees in the forest is the raw importance score for the variable m.

For each case, consider all the trees for which it is oob. Subtract the percentage of votes for the correct class in the variable-m-permuted oob data from the percentage of votes for the correct class in the untouched oob data. This is the local importance score for variable m for this case, and is used in the graphics program RAFT (Random Forest Tool).

For further details on random forests, please refer to the random forests website http://www.math.usu.edu/~adele/forests/cc_home.htm maintained by Professor Adele Cutler at Utah State University.

In summary, random forests do not overfit and and enjoy prediction accuracy that is as good as AdaBoost and sometimes better. The algorithm runs fast on large high-dimensional data and is somewhat robust to outliers. It also has an effective mechanism for handling missing data. In the forest-building process, it internally estimates the classification error, the strength of each tree and the correlation between trees. It also distinguishes itself from some black-box methods (e.g. neural networks) by providing the importance score for each predictor, and hence makes the output more interpretable to users. Furthermore, random forests can serve as an exploratory tool to find interactions among predictors, locate outliers and provide interesting views of the data. Its application can also be extended to unsupervised clustering.

30.5 Conclusion

In this chapter, we have discussed several tree-based methods for classification and regression. Of course, many more supervised learning methods are available, including multivariant adaptive regression splines (MARS), neural networks, and support vector machines (SVM). In this last section, we discuss the relative merits of tree-based methods among this much larger set of well-known supervised learning tools.

Hastie et al. [30.12] note that typical characteristics found in real-world data sets make direct application of most supervised-learning tools difficult. First, data-mining applications tend to involve very large data sets in terms of both the number of observations and the number of variables (the majority of which are often irrelevant). Moreover, these data sets generally contain both quantitative and qualitative variables. The quantitative variables are typically measured on different scales, and the qualitative variables may have different numbers of categories. Missing data are abundant, and outliers are also very common.

Tree-based methods are particularly well-suited to deal with these difficulties. Trees grow quickly, so the

size of a data set is not a big concern. Tree algorithms readily admit mixed variable types, and feature selection is a part of the building process, so irrelevant variables have little impact on the resulting model. Tree-building methods account for missing data in an effective way, and the results for classification or prediction are often robust against outliers.

Many other supervised learning methods fall short in some of these areas. MARS has difficulty with outliers in predictor variables, and transformations on variables can dramatically impact its results. Neural networks and SVM both require dummy coding of categorical variables, they are not adept at handling missing values, and they are sensitive to outliers and transformations.

Tree-based methods have one other important advantage over black-box techniques such as neural networks; tree models are much more readily interpretable. This characteristic is vital to those applications for which predictive accuracy is secondary to the main goal of obtaining qualitative insight into the structure of the data.

In spite of these advantages, tree-based methods do suffer one key drawback: a relative lack of predictive power. Neural networks and support vector machines commonly outperform classification and regression trees, particularly when the underlying structure of the data depends on linear combinations of variables. As we discussed in Sect. 30.4, ensemble methods such as boosting and random forests can be quite effective at improving their accuracy. However, this predictive improvement comes with some cost. Ensemble methods lose the interpretive value in a single tree, and they are much more computationally expensive. The tree-based methods do not always yield the best possible results for classification and prediction, but they are worth a try in a wide variety of applications. In any scientific application, we certainly encourage you to see the forest – not just a few trees.

References

30.1 C. L. Blake, C. J. Merz: *UCI repository of machine learning databases http://www.ics.uci.edu/mlearn/MLRepository.html* (Department of Information and Computer Science (Univ. California), Irvine 1998)

30.2 K.-Y. Chan, W.-Y. Loh: LOTUS: An algorithm for building accurate, comprehensible logistic regression trees, J. Comput. Graph. Stat. **13**(4), 826–852 (2004)

30.3 Federal Emergency Management Agency: *Typical Costs of Seismic Rehabilitation of Existing Buildings*, FEMA 156, Vol. 1–Summary, 2nd edn. (FEMA, Washington 1993)

30.4 Federal Emergency Management Agency: *Typical Costs of Seismic Rehabilitation of Existing Buildings*, FEMA 157, Vol. 2–Supporting Documentation, 2nd edn. (FEMA, Washington 1993)

30.5 L. Breiman, J. Friedman, R. Olshen, C. Stone: *Classification and Regression Trees* (Chapman Hall, New York 1984)

30.6 W.-Y. Loh, Y.-S. Shih: Split selection methods for classificaiton trees, Stat. Sin. **7**, 815–840 (1997)

30.7 H. Kim, W.-Y. Loh: Classification trees with unbiased multiway splits, J. Am. Stat. Assoc. **96**, 589–604 (2001)

30.8 W.-Y. Loh: Regression trees with unbiased variable selection, interaction detection, Stat. Sin. **12**, 361–386 (2002)

30.9 J. R. Quinlan: *C4.5: Programs for Machine Learning* (Morgan Kaufmann, San Mateo 1993)

30.10 G. V. Kass: An exploratory technique for investigating large quantities of categorical data, Appl. Stat. **29**, 119–127 (1980)

30.11 R. A. Fisher: The use of multiple measurements in taxonomic problems, Ann. Eugenic. **7**, 179–188 (1936)

30.12 T. Hastie, R. Tibshirani, J. Friedman: *The Elements of Statistical Learning: Data Mining, Inference, Prediction* (Springer, Berlin Heidelberg New York 2001)

30.13 F. Esposito, D. Malerba, G. Semeraro: A comparative analysis of methods for pruning decision trees, IEEE Trans. Pattern Anal. **19**, 476–491 (1997)

30.14 P. Ein-Dor, J. Feldmesser: Attributes of the performance of central processing units: a relative performance prediction model, Commun. ACM **30**, 308–317 (1987)

30.15 R. J. Little, D. B. Rubin: *Statistical Analysis with Missing Data*, 2nd edn. (Wiley, Boboken 2002)

30.16 L. Breiman: Bagging predictors, Mach. Learn. **24**, 123–140 (1996)

30.17 H. Drucker, C. Cortes: Boosting decision trees. In: *Adv. Neur. Inf. Proc. Syst.*, Proc. NIPS'95, Vol. 8, ed. by M. C. Mozer D. S. Touretzky, E. Hasselmo (Ed.) M. (MIT Press, Cambridge 1996) pp. 479–485

30.18 W.-Y. Loh, N. Vanichsetakul: Tree-structured classification via generalized discriminant analysis (with discussion), J. Am. Stat. Assoc. **83**, 715–728 (1988)

30.19 J. A. Hartigan, M. A. Wong: Algorithm 136, A *k*-means clustering algorithm, Appl. Stat. **28**, 100 (1979)
30.20 J. R. Quinlan: Discovering rules by induction from large collections of examples. In: *Expert Systems in the Micro-Electronic Age*, ed. by D. Michie (Edinburgh Univ. Press, Edinburgh 1979) pp. 168–201
30.21 E. B. Hunt, J. Marin, P. J. Stone: *Experiments in Induction* (Academic, New York 1966)
30.22 J. Dougherty, R. Kohavi, M. Sahami: Supervised, unsupervised discretization of continuous features. In: *Proceedings of the Twelfth International Conference on Machine Learning*, ed. by A. Prieditis, S. J. Russel (Morgan Kaufmann, San Mateo 1995) pp. 194–202
30.23 J. R. Quinlan: Improved use of continuous attributes in C4.5, J. Artif. Intell. Res. **4**, 77–90 (1996)
30.24 T.-S. Lim, W.-Y. Loh, Y.-S. Shih: A comparison of prediction accuracy, complexity, training time of thirty-three old and new classification algorithms, Mach. Learn. J. **40**, 203–228 (2000)
30.25 E. Bauer, R. Kohavi: An empirical comparison of voting classification algorithms: bagging, boosting, variants, Mach. Learn. **36**, 105–139 (1999)
30.26 L. Breiman: Statistical modeling: the two cultures, Stat. Sci. **16**, 199–215 (2001)
30.27 L. Breiman: Random forests, Mach. Learn. **45**, 5–32 (2001)
30.28 T. G. Dietterich: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, randomization, Mach. Learn. **40**, 139–157 (2000)
30.29 Y. Freund, R. E. Schapire: Experiments with a new boosting algorithm. In: *Machine Learning: Proceedings of the Thirteenth International Conference*, ed. by L. Saitta (Morgan Kaufmann, San Mateo 1996) pp. 148–156
30.30 R. Schapire: The strength of weak learnability, Mach. Learn. **5**(2), 197–227 (1990)
30.31 Y. Freund: Boosting aweak learning algorithm by majority, Inform. Comput. **121**(2), 256–285 (1995)
30.32 Y. Freund, R. E. Schapire: A decision-theoretic generalization of on-line learning, an application to boosting, J. Comput. Syst. Sci. **55**, 119–139 (1997)
30.33 J. Friedman, T. Hastie, R. Tibshirani: Additive logistic regression: astatistical view of boosting (with discussion), Ann. Stat. **28**, 337–374 (2000)
30.34 T. K. Ho: The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. **20**, 832–844 (1998)
30.35 M. R. Segal: *Machine learning benchmarks, random forest regression*, Technical Report, Center for Bioinformatics and Molecular Biostatistics (Univ. California, San Francisco 2004)

31. Image Registration and Unknown Coordinate Systems

This chapter deals with statistical problems involving unknown coordinate systems, either in Euclidean 3-space $\mathbb{R}^3$ or on the unit sphere Ω_3 in $\mathbb{R}^3$. We also consider the simpler cases of Euclidean 2-space $\mathbb{R}^2$ and the unit circle Ω_2. The chapter has five major sections.

Although other problems of unknown coordinate systems have arisen, a very important problem of this class is the problem of *image registration* from landmark data. In this problem we have two images of the same object (such as satellite images taken at different times) or an image of a prototypical object and an actual object. It is desired to find the rotation, translation, and possibly scale change, which will best align the two images. Whereas many problems of this type are two-dimensional, it should be noted that medical imaging is often three-dimensional.

After introducing some mathematical preliminaries we introduce the concept of M-*estimators*, a generalization of least squares estimation. In least squares estimation, the registration that minimizes the sum of squares of the lengths of the deviations is chosen; in M estimation, the sum of squares of the lengths of the deviations is replaced by some other objective function. An important case is L_1 estimation, which minimizes the sum of the lengths of the deviations; L_1 estimation is often used when the possibility of outliers in the data is suspected.

The second section of this chapter deals with the calculation of least squares estimates. Then, in the third section, we introduce an iterative modification of the least squares algorithm to calculate other M-estimates. Note that minimization usually involves some form of differentiation and hence this section starts with a short introduction to the geometry of the group of rotations and differentiation in the rotation group. Many statistical techniques are based upon approximation by derivatives and hence a little understanding of geometry is necessary to understand the later statistical sections.

The fourth section discusses the statistical properties of M-estimates. A great deal of emphasis is placed upon the relationship between the geometric configuration of the landmarks and the statistical errors in the image registration. It is shown that these statistical errors are determined, up to a constant, by the geometry of the landmarks. The constant of proportionality depends upon the objective function and the distribution of the errors in the data.

General statistical theory indicates that, if the data error distribution is (anisotropic) multivariate normal, least squares estimation is optimal. An important result of this section is that, even in this case when least squares estimation is theoretically the most efficient, the use of L_1 estimation can guard against outliers with a very modest cost in efficiency. Here optimality and efficiency refer to the expected size of the statistical errors. In practice, data is often long-tailed and L_1 estimation yields *smaller* statistical errors than least squares estimation. This will be the case with the three-dimensional image registration example given here.

Finally, in the fifth section, we discuss diagnostics that can be used to determine which data points are most influential upon the registration. Thus, if the registration is unsatisfactory, these diagnostics can be used to determine which data points are most responsible and should be reexamined.

31.1 Unknown Coordinate Systems and Their Estimation

31.1.1 Problems of Unknown Coordinate Systems

Wahba [31.1] posed the following question. Suppose we have the directions of certain stars with respect to the unknown coordinate system of a satellite. How can we estimate the orientation of the satellite? Let $\mathbf{A}$ be the unknown 3×3 matrix whose rows represent the axes of the satellite's coordinate system with respect to a fixed and known (Earth) coordinate system. Furthermore let $\boldsymbol{u}_i$ be the directions of the stars with respect to the known coordinate systems, where each $\boldsymbol{u}_i$ is written as a three-dimensional column vector with unit length. Similarly let $\boldsymbol{v}_i$ be the directions of the stars with respect to the satellite's coordinate system. Then

$$\boldsymbol{v}_i = \mathbf{A}\boldsymbol{u}_i + \text{error}\,. \tag{31.1}$$

In essence the question was to estimate $\mathbf{A}$. Wahba gave the least squares solution.

Chapman et al. [31.2] posed the same question in the following form. Suppose we have an object defined by a computer-aided design (CAD) program and a prototype is measured using a coordinate measuring machine (CMM). The orientations of lines on the object can be defined by unit vectors parallel to the lines and the orientations of planes can be defined by unit vectors normal to the planes. So we have unit vectors $\boldsymbol{u}_i$ defined by the CAD program and the corresponding unit vectors $\boldsymbol{v}_i$ as measured by the CMM. If $\mathbf{A}$ is the coordinate system of the CMM relative to the CAD program, then (31.1) holds.

Chapman et al. again used a least squares estimate $\widehat{\mathbf{A}}$ of $\mathbf{A}$. The main question of interest, that is the geometric integrity of the prototype, was then answered by analyzing the residuals of $\boldsymbol{v}_i$ from $\widehat{\mathbf{A}}\boldsymbol{u}_i$.

Since the $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are of unit length, these two problems involve spherical data.

31.1.2 Image Registration

If we enlarge the inquiry to Euclidean space data, we arrive at the widely used *image registration* problem. Suppose $\boldsymbol{u}_i \in \mathbb{R}^p$ represent the locations of some landmarks in one image, and $\boldsymbol{v}_i \in \mathbb{R}^p$ the locations of corresponding landmarks in a second image of the same

object. The usual applications occur with $p = 2.3$. Under certain conditions, it might be reasonable to suppose that

$$\boldsymbol{v}_i = \mathbf{B}\boldsymbol{u}_i + \boldsymbol{b} + \text{error} \tag{31.2}$$

for an unknown $p \times p$ matrix $\mathbf{B}$ and an unknown p-dimensional column vector $\boldsymbol{b}$. The matrix $\mathbf{B}$ represents a coordinate change and the vector $\boldsymbol{b}$ represents a translation of coordinates. The image registration problem is to estimate $\mathbf{B}$ and $\boldsymbol{b}$.

The model (31.2) also arises in a slightly different context. Suppose we have landmarks $\boldsymbol{u}_i$ on a prototypical face. For example the $\boldsymbol{u}_i$ might represent the locations of the nose, the two eyes, the base of the chin, etc. For the purpose of automated processing of a large number of facial images of different subjects, we might want to bring each facial image into alignment with the prototypical image using a transformation of the form (31.2) where the $\boldsymbol{v}_i$ represent the same locations (nose, two eyes, base of chin, etc.) on the subject facial image.

In the absence of measurement error, one does not expect the landmarks on two faces to be related using a transformation of the form

$$\boldsymbol{v}_i = \mathbf{B}\boldsymbol{u}_i + \boldsymbol{b} \,. \tag{31.3}$$

The reader might be puzzled why a transformation of this form is under consideration. Statistical error, however, is not limited to measurement error. Statistical error incorporates all effects not included in the systematic portion of the model. In building a model of the form (31.2), we hope to separate out the most important relationship (31.3) between the landmarks $\boldsymbol{u}_i$ on one object and the corresponding landmarks $\boldsymbol{v}_i$ on the other object; the rest is placed in the statistical error.

Unlike the Wahba problem, the unknown $(\mathbf{B}, \boldsymbol{b})$ of the image registration problem, or the unknown $\mathbf{A}$ in the *Chapman* et al. problem, are not of primary interest. Rather, they must be estimated as a preliminary step to more interesting problems. We will discuss herein the properties of various methods of estimating these unknowns. These properties will hopefully help the interested reader to choose a good estimation technique which will hopefully yield better results after this preliminary step is completed.

31.1.3 The Orthogonal and Special Orthogonal Matrices

Consider, for example the data set in Table 31.1 from *Chang* and *Ko* [31.3], which we will analyze repeatedly in what follows. This data consists of the digitized locations of 12 pairs of landmarks on the left and right hands of one of the authors. This is a $p = 3$ three-dimensional image registration problem. We might decide that, apart from the statistical error term, the shape of the two hands is the same; that is the distance between two points on one hand is the same as the distance between the corresponding two points on the other hand.

This condition translates mathematically to the equation $\mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{I}_p$, the $p \times p$-dimensional identity matrix. We outline a derivation of this well-known mathematical fact for the primary purpose of introducing the reader to the mathematical style of the remainder of this chapter. The distance between two p-dimensional column vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ is

$$||\boldsymbol{v}_2 - \boldsymbol{v}_1|| = \sqrt{(\boldsymbol{v}_2 - \boldsymbol{v}_1)^{\mathrm{T}}(\boldsymbol{v}_2 - \boldsymbol{v}_1)} \,, \tag{31.4}$$

where the operations on the right-hand side of (31.4) are matrix multiplication and transposition. If the $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$ are related by (31.3),

$$\begin{aligned}\left(\boldsymbol{v}_j - \boldsymbol{v}_i\right)^{\mathrm{T}} \left(\boldsymbol{v}_j - \boldsymbol{v}_i\right) &= \left[\mathbf{B}(\boldsymbol{u}_j - \boldsymbol{u}_i)\right]^{\mathrm{T}} \times \left[\mathbf{B}(\boldsymbol{u}_j - \boldsymbol{u}_i)\right] \\ &= \left[(\boldsymbol{u}_j - \boldsymbol{u}_i)\right] \mathbf{B}^{\mathrm{T}}\mathbf{B} \left[(\boldsymbol{u}_j - \boldsymbol{u}_i)\right] \,.\end{aligned}$$

Thus if $||\boldsymbol{v}_j - \boldsymbol{v}_i|| = ||\boldsymbol{u}_j - \boldsymbol{u}_i||$ for all i and j, and if the $\boldsymbol{u}_i$ do not all lie in a $(p-1)$-dimensional hyperplane of $\mathbb{R}^p$,

$$\mathbf{I}_p = \mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{B}\mathbf{B}^{\mathrm{T}} \,. \tag{31.5}$$

Table 31.1 12 digitized locations on the left and right hand

	Left hand $\boldsymbol{u}_i$			Right hand $\boldsymbol{v}_i$		
A	5.17	11.30	16.18	5.91	11.16	16.55
B	7.40	12.36	17.50	8.63	10.62	18.33
C	8.56	12.59	17.87	10.09	10.60	18.64
D	9.75	13.62	17.01	10.89	10.95	17.90
E	11.46	14.55	12.96	12.97	10.13	13.88
F	7.10	13.12	12.56	8.79	11.21	13.17
G	8.85	13.82	12.60	10.70	11.10	13.42
H	6.77	13.07	10.32	8.47	11.09	11.35
I	6.26	11.62	13.34	7.28	12.52	14.04
J	6.83	12.00	13.83	8.05	12.42	14.56
K	7.94	12.29	13.84	9.07	12.39	14.86
L	8.68	12.71	13.67	10.15	12.17	14.44

A: Top of little finger; B: Top of ring finger; C: Top of middle finger; D: Top of forefinger; E: Top of thumb; F: Gap between thumb and forefinger; G: Center of palm; H: Base of palm; I: Little finger knuckle; J: Ring finger knuckle; K: Middle finger knuckle; L: Forefinger knuckle

Note that the first equality of (31.5) implies that $\mathbf{B}^{-1} = \mathbf{B}^{\mathrm{T}}$ and hence the second equality follows. Matrices which satisfy condition (31.5) are said to be *orthogonal*.

On the other hand, we might want to hypothesize that the two hands (again apart from statistical error) have the same shape except that one hand might be larger than the other. In this case we are hypothesizing

$$\mathbf{B} = \gamma \mathbf{A} \tag{31.6}$$

where $\mathbf{A}$ is orthogonal and γ is a positive real number.

In the Wahba and Chapman et al. problems, the rows of $\mathbf{A}$ are known to be an orthonormal basis of $\mathbb{R}^3$. Since the (i, j) entry of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ is the dot product of the i-th and j-th rows of $\mathbf{A}$, it follows that $\mathbf{A}$ is orthogonal. However, more is known. Since the unknown coordinate system is known to be right-handed,

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{I}_p, \quad \det(\mathbf{A}) = 1\,, \tag{31.7}$$

where $\det(\mathbf{A})$ is the determinant of the matrix $\mathbf{A}$. Such matrices are said to be *special orthogonal*.

In the hands data of Table 31.1, if we use the model (31.2) with condition (31.6), then $\mathbf{A}$ will not be special orthogonal. This is because the left and right hands have different orientations. However, it is common in image registration problems to assume that condition (31.6) is true with $\mathbf{A}$ assumed to be special orthogonal.

Following standard mathematical notation, we will use $\mathcal{O}(p)$ to denote the $p \times p$ orthogonal matrices [that is the set of all matrices which satisfy (31.5) and $\mathcal{SO}(p)$ to denote the subset of $\mathcal{O}(p)$ of special orthogonal matrices [that is the set of all matrices which satisfy (31.7)].

31.1.4 The Procrustes and Spherical Regression Models

In this chapter, we will be concerned with statistical methods which apply to the model (31.2) for Euclidean space data $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^p$, for arbitrary p, where $\mathbf{B}$ satisfies the condition (31.6) with $\mathbf{A}$ constrained to be either orthogonal or special orthogonal. Following Goodall [31.4], we will call this model the *Procrustes* model.

We will also consider models of the form (31.1), where the p-vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are constrained to be of unit length, that is

$$\boldsymbol{u}_i, \boldsymbol{v}_i \in \Omega_p = S^{p-1} = \left(\boldsymbol{x} \in \mathbb{R}^p \;\middle|\; \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1\right)$$

and $\mathbf{A}$ is constrained to be either orthogonal or special orthogonal. Following *Chang* [31.5], we will call this model the *spherical regression* model.

The statistical methodology for these two models can easily be described in parallel. In general, we will focus on the Procrustes model, while giving the modifications that apply to the spherical regression model.

31.1.5 Least Squares, L_1, and M Estimation

In Sect. 31.2, we will derive the least squares estimate of $\mathbf{A}$, γ, $\boldsymbol{b}$ for the Procrustes model. This estimate minimizes

$$\rho_2(\mathbf{A}, \gamma, \boldsymbol{b}) = \sum_i ||\boldsymbol{v}_i - \gamma \mathbf{A}\boldsymbol{u}_i - \boldsymbol{b}||^2 \tag{31.8}$$

over all $\mathbf{A}$ in either $\mathcal{O}(p)$ or $\mathcal{SO}(p)$, constants $\gamma > 0$, and p-vectors $\boldsymbol{b} \in \mathbb{R}^p$. For the spherical regression model, the least squares estimate minimizes

$$\rho_2(\mathbf{A}) = \sum_i ||\boldsymbol{v}_i - \mathbf{A}\boldsymbol{u}_i||^2 \tag{31.9}$$

$$= 2n - 2\sum_i \boldsymbol{v}_i^{\mathrm{T}}\mathbf{A}\boldsymbol{u}_i \tag{31.10}$$

over all $\mathbf{A}$ in either $\mathcal{O}(p)$ or $\mathcal{SO}(p)$. For the second equality in (31.9), we have used that if $1 = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}$, then

$$\begin{aligned} ||\boldsymbol{v} - \mathbf{A}\boldsymbol{u}||^2 &= (\boldsymbol{v} - \mathbf{A}\boldsymbol{u})^{\mathrm{T}}\,(\boldsymbol{v} - \mathbf{A}\boldsymbol{u}) \\ &= \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{v}^{\mathrm{T}}\mathbf{A}\boldsymbol{u} - (\mathbf{A}\boldsymbol{u})^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{u}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\boldsymbol{u} \\ &= 2 - 2\boldsymbol{v}^{\mathrm{T}}\mathbf{A}\boldsymbol{u}\,. \end{aligned}$$

Least squares estimates have the advantage that an explicit closed-form solution for them is available. They have the disadvantage that they are very sensitive to *outliers*, that is points $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ for which the error term in (31.2) is unusually large. In the image registration problem, an outlier can arise in several contexts. It can be the result of a measurement error, or it can be the result of a misidentified landmark. Perhaps the image is not very clear, or the landmark (e.g. 'point of the nose') cannot be very precisely determined, or the landmark is obscured (by clouds or shrubs, etc.). Or perhaps there are places in the image where the image is not really rigid, that is the ideal match (31.3) does not apply very well. It is easy to conceive of a myriad of situations which might give rise to outliers.

L_1 estimators are often used to ameliorate the effects of outliers. These estimators minimize

$$\rho_1(\mathbf{A}, \gamma, \boldsymbol{b}) = \sum_i ||\boldsymbol{v}_i - \gamma \mathbf{A}\boldsymbol{u}_i - \boldsymbol{b}||\,, \tag{31.11}$$

for the Procrustes model, or the sum of the distances along the surface of the sphere

$$\rho_1(\mathbf{A}) = \sum_i \arccos(\boldsymbol{v}_i^{\mathrm{T}} \mathbf{A} \boldsymbol{u}_i) \tag{31.12}$$

for the spherical regression model. Unfortunately, an explicit closed-form solution for the L_1 estimate is not available and it must be calculated by numerical minimization. We will offer a few suggestions on approaches for numerical minimization in Sect. 31.3.5.

The least squares and L_1 estimators are special cases of the so-called *M estimators*. These estimators minimize an objective function of the form

$$\rho(\mathbf{A}, \gamma, \boldsymbol{b}) = \sum_i \rho_0(s_i) \,, \tag{31.13}$$

where

$$s_i = ||\boldsymbol{v}_i - \gamma \mathbf{A} \boldsymbol{u}_i - \boldsymbol{b}||$$

and ρ_0 is some increasing function. Intermediate between the least squares and L_1 estimate is the *Huber* estimate for which

$$\rho_0(s) = \begin{cases} (s/b)^2 & s < b \\ s/b & s \geq b \end{cases}$$

for some preset constant b. Or we can *Windsorize* the estimate

$$\rho_0(s) = \begin{cases} (s/b)^2 & s < b \\ 1 & s \geq b \end{cases} .$$

In the linear regression context, these and other objective functions are discussed in *Huber* [31.6].

For the spherical regression model, an M-estimator minimizes an objective function of the form

$$\rho(\mathbf{A}) = \sum_i \rho_0(t_i) \,, \tag{31.14}$$

where

$$t_i = \boldsymbol{v}_i^{\mathrm{T}} \mathbf{A} \boldsymbol{u}_i \,.$$

Notice that, as $\boldsymbol{v}$ moves away from $\mathbf{A}\boldsymbol{u}$ towards the antipodal point $-\mathbf{A}\boldsymbol{u}$, $t = \boldsymbol{v}^{\mathrm{T}} \mathbf{A} \boldsymbol{u}$ *decreases* from 1 to -1. Thus, for the spherical case, $\rho_0(t)$ is chosen to be a decreasing function of t.

In Sect. 31.4 we will discuss the statistical properties of M-estimates. We will see how the geometry of the data translates into the error structure of the estimate. In the image registration problem, this information can be used, for example, to help select landmarks. General statistical theory indicates under certain conditions ("normal distribution") the least squares solution is optimal. However, if we were to use a L_1 estimate to guard against outliers, we would suffer a penalty of 13% for image registrations in two dimensions and only 8% for image registrations in three dimensions, even when least squares is theoretically optimal. We will make more precise in Sect. 31.4 how this *penalty* is defined. The important point to realize is that, especially for three-dimensional image registrations, L_1 estimators offer important protections against outliers in the data at very modest cost in the statistical efficiency of the estimator.

In Sect. 31.5, we will discuss diagnostics for the Procrustes and spherical regression models. If the image registration is not satisfactory, this section will give tools to determine which of the landmarks is causing the unsatisfactory registration. It will follow, for example, that landmarks which greatly influence $\mathbf{A}$ will have negligible influence on γ and vice versa.

31.2 Least Squares Estimation

31.2.1 Group Properties of $\mathcal{O}(p)$ and $\mathcal{SO}(p)$

It is important to note that $\mathcal{O}(p)$ and $\mathcal{SO}(p)$ are groups in the mathematical sense. That is, if $\mathbf{A}, \mathbf{B} \in \mathcal{O}(p)$, then

$$(\mathbf{AB})^{\mathrm{T}}(\mathbf{AB}) = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{AB} = \mathbf{B}^{\mathrm{T}}\mathbf{I}_p\mathbf{B} = \mathbf{I}_p$$

since both $\mathbf{A}$ and $\mathbf{B}$ satisfy (31.5). Thus $\mathbf{AB} \in \mathcal{O}(p)$. Similarly if $\mathbf{A} \in \mathcal{O}(p)$, then (31.5) implies that $\mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}} \in \mathcal{O}(p)$. This implies that $\mathcal{O}(p)$ is a group. Furthermore, if $\det(\mathbf{A}) = \det(\mathbf{B}) = 1$, then $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B}) = 1$ and $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A}) = 1$. In summary we have

$$\begin{aligned} &\text{If } \mathbf{A}, \mathbf{B} \in \mathcal{O}(p), \text{ then } \mathbf{AB} \in \mathcal{O}(p) \\ &\text{and } \mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}} \in \mathcal{O}(p) \\ &\text{If } \mathbf{A}, \mathbf{B} \in \mathcal{SO}(p), \text{ then } \mathbf{AB} \in \mathcal{SO}(p) \\ &\text{and } \mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}} \in \mathcal{SO}(p) \,. \end{aligned} \tag{31.15}$$

Notice also that, if $\mathbf{A}$ satisfies (31.5), then

$$1 = \det(\mathbf{A}^{\mathrm{T}}\mathbf{A}) = [\det(\mathbf{A})]^2$$

so that $\det(\mathbf{A}) = 1, -1$.

31.2.2 Singular Value Decomposition

Given a $p \times q$ matrix $\mathbf{X}$ its *singular value decomposition* is

$$\mathbf{X} = \mathbf{O}_1 \Lambda \mathbf{O}_2^{\mathrm{T}} , \tag{31.16}$$

where $\mathbf{O}_1 \in \mathcal{O}(p)$, $\mathbf{O}_2 \in \mathcal{O}(q)$ and $\mathbf{\Lambda}$ is $p \times q$. If $p \le q$, $\mathbf{\Lambda}$ has block form

$$\mathbf{\Lambda} = \left[\operatorname{diag}(\lambda_1, \cdots, \lambda_p) \;\; \mathbf{0}_{(p,q-p)} \right]$$

Here $\operatorname{diag}(\lambda_1, \cdots, \lambda_p)$ is a diagonal matrix with entries $\lambda_1 \ge \cdots \ge \lambda_p$ and $\mathbf{0}_{(p,q-p)}$ is a $p \times (q-p)$ matrix with all zeros. If $q \le p$

$$\mathbf{\Lambda} = \begin{pmatrix} \operatorname{diag}(\lambda_1, \cdots, \lambda_q) \\ \mathbf{0}_{(p-q,q)} \end{pmatrix} .$$

Most mathematical software packages now include the singular value decomposition. However, it can be computed using a package which only computes eigen-decompositions of symmetric matrices. Suppose temporarily $p \le q$. Since $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ is a symmetric nonnegative definite matrix, its eigen-decomposition has the form

$$\mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{O}_1 \mathbf{\Lambda}_1 \mathbf{O}_1^{\mathrm{T}} ,$$

where $\mathbf{O}_1 \in \mathcal{O}(p)$ and $\mathbf{\Lambda}_1 = \operatorname{diag}(\lambda_1^2, \cdots, \lambda_p^2)$ with $\lambda_1 \ge \cdots \ge \lambda_p \ge 0$. The columns of $\mathbf{O}_1$ are the eigenvectors of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ and $\lambda_1^2, \cdots, \lambda_p^2$ are the corresponding eigenvalues. Suppose $\lambda_p > 0$ and let $\widetilde{\mathbf{O}}_2 = \mathbf{X}^{\mathrm{T}} \mathbf{O}_1 \mathbf{\Lambda}_1^{-1/2}$. $\widetilde{\mathbf{O}}_2$ is $q \times p$, but

$$\begin{aligned} \widetilde{\mathbf{O}}_2^{\mathrm{T}} \widetilde{\mathbf{O}}_2 &= \mathbf{\Lambda}_1^{-1/2} \mathbf{O}_1^{\mathrm{T}} \mathbf{X}\mathbf{X}^{\mathrm{T}} \mathbf{O}_1 \mathbf{\Lambda}_1^{-1/2} \\ &= \mathbf{\Lambda}_1^{-1/2} \mathbf{O}_1^{\mathrm{T}} \mathbf{O}_1 \mathbf{\Lambda}_1 \mathbf{O}_1^{\mathrm{T}} \mathbf{O}_1 \mathbf{\Lambda}_1^{-1/2} \\ &= \mathbf{\Lambda}_1^{-1/2} \mathbf{\Lambda}_1 \mathbf{\Lambda}_1^{-1/2} = \mathbf{I}_p , \end{aligned}$$

so that the columns of $\widetilde{\mathbf{O}}_2$ are orthonormal. Furthermore

$$\mathbf{O}_1 \mathbf{\Lambda}_1^{1/2} \widetilde{\mathbf{O}}_2^{\mathrm{T}} = \mathbf{O}_1 \mathbf{\Lambda}_1^{1/2} \mathbf{\Lambda}_1^{-1/2} \mathbf{O}_1^{\mathrm{T}} \mathbf{X} = \mathbf{X} .$$

Filling $\mathbf{\Lambda}_1^{1/2}$ with $q - p$ columns of zeros, and completing the columns of $\widetilde{\mathbf{O}}_2$ to an orthonormal basis of $\mathbb{R}^q$ yields the decomposition (31.16).

Extensions to the cases when $\lambda_p = 0$ or when $q \le p$ will not be difficult for the careful reader.

31.2.3 Least Squares Estimation in the Procrustes Model

The least squares estimation of the Procrustes model (31.2) has long been known (see, for example, Goodall [31.4]). Let $\overline{\boldsymbol{u}} = n^{-1} \sum_i \boldsymbol{u}_i$, where n is the number of pairs $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ and let $\overline{\boldsymbol{v}}$ be similarly defined. Define the $p \times p$ matrix $\mathbf{X}$ by

$$\mathbf{X} = \sum_i (\boldsymbol{u}_i - \overline{\boldsymbol{u}})(\boldsymbol{v}_i - \overline{\boldsymbol{v}})^{\mathrm{T}} .$$

Then

$$\begin{aligned} \rho_2(\mathbf{A}, \gamma, \boldsymbol{b}) &= \sum_i ||\boldsymbol{v}_i - \gamma \mathbf{A}\boldsymbol{u}_i - \boldsymbol{b}||^2 \\ &= \sum_i ||\boldsymbol{v}_i - \overline{\boldsymbol{v}} - \gamma \mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \\ &\quad - [\boldsymbol{b} - (\overline{\boldsymbol{v}} - \gamma \mathbf{A}\overline{\boldsymbol{u}})]\,||^2 \\ &= \sum_i ||\boldsymbol{v}_i - \overline{\boldsymbol{v}}||^2 \\ &\quad - \gamma \sum_i (\boldsymbol{u}_i - \overline{\boldsymbol{u}})^{\mathrm{T}} \mathbf{A}^{\mathrm{T}} (\boldsymbol{v}_i - \overline{\boldsymbol{v}}) \\ &\quad - \gamma \sum_i (\boldsymbol{v}_i - \overline{\boldsymbol{v}})^{\mathrm{T}} \mathbf{A} (\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \\ &\quad + \gamma^2 \sum_i ||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||^2 \\ &\quad + n||\boldsymbol{b} - (\overline{\boldsymbol{v}} - \gamma \mathbf{A}\overline{\boldsymbol{u}})||^2 . \end{aligned}$$

All the other cross-product terms sum to zero. Now

$$\begin{aligned} &\sum_i (\boldsymbol{v}_i - \overline{\boldsymbol{v}})^{\mathrm{T}} \mathbf{A} (\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \\ &= \sum_i \operatorname{Tr}\left[(\boldsymbol{v}_i - \overline{\boldsymbol{v}})^{\mathrm{T}} \mathbf{A} (\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \right] \\ &= \sum_i \operatorname{Tr}\left[\mathbf{A} (\boldsymbol{u}_i - \overline{\boldsymbol{u}})(\boldsymbol{v}_i - \overline{\boldsymbol{v}})^{\mathrm{T}} \right] = \operatorname{Tr}(\mathbf{A}\mathbf{X}) \end{aligned}$$

and

$$\begin{aligned} &\sum_i (\boldsymbol{u}_i - \overline{\boldsymbol{u}})^{\mathrm{T}} \mathbf{A}^{\mathrm{T}} (\boldsymbol{v}_i - \overline{\boldsymbol{v}}) \\ &= \sum_i (\boldsymbol{v}_i - \overline{\boldsymbol{v}})^{\mathrm{T}} \mathbf{A} (\boldsymbol{u}_i - \overline{\boldsymbol{u}}) = \operatorname{Tr}(\mathbf{A}\mathbf{X}) . \end{aligned}$$

Therefore

$$\begin{aligned} \rho_2(\mathbf{A}, \gamma, \boldsymbol{b}) &= \sum_i ||\boldsymbol{v}_i - \overline{\boldsymbol{v}}||^2 - 2\gamma \operatorname{Tr}(\mathbf{A}\mathbf{X}) \\ &\quad + \gamma^2 \sum_i ||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||^2 \\ &\quad + n||\boldsymbol{b} - (\overline{\boldsymbol{v}} - \gamma \mathbf{A}\overline{\boldsymbol{u}})||^2 . \end{aligned} \tag{31.17}$$

Substituting (31.16),

$$\begin{aligned} \operatorname{Tr}(\mathbf{A}\mathbf{X}) &= \operatorname{Tr}\left(\mathbf{A}\mathbf{O}_1 \mathbf{\Lambda} \mathbf{O}_2^{\mathrm{T}} \right) = \operatorname{Tr}\left(\mathbf{O}_2^{\mathrm{T}} \mathbf{A}\mathbf{O}_1 \mathbf{\Lambda} \right) \\ &= \sum_i \lambda_i e_{ii} , \end{aligned}$$

where e_{ii} are the diagonal entries of $\mathbf{O}_2^{\mathrm{T}}\mathbf{A}\mathbf{O}_1 \in \mathcal{O}(p)$. Now $|e_{ii}| \le 1$ and hence $\mathrm{Tr}\,(\mathbf{A}\mathbf{X})$ is maximized when $e_{ii} = 1$ or, equivalently, when $\mathbf{O}_2^{\mathrm{T}}\mathbf{A}\mathbf{O}_1 = \mathbf{I}_p$. This implies $\mathbf{A} = \mathbf{O}_2\mathbf{O}_1^{\mathrm{T}}$.

Thus if $\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}}\right)$ minimizes (31.17),

$$\begin{aligned}
\widehat{\mathbf{A}} &= \mathbf{O}_2\mathbf{O}_1^{\mathrm{T}}\,, \\
\widehat{\gamma} &= \left(\sum_i ||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||^2\right)^{-1} \mathrm{Tr}\left(\widehat{\mathbf{A}}\mathbf{X}\right) \\
&= \left(\sum_i ||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||^2\right)^{-1} \sum_i \lambda_i\,, \qquad (31.18) \\
\widehat{\boldsymbol{b}} &= \overline{\boldsymbol{v}} - \widehat{\gamma}\widehat{\mathbf{A}}\overline{\boldsymbol{u}}\,.
\end{aligned}$$

If $\mathbf{A}$ is constrained to lie in $\mathcal{SO}(p)$, we use a modified singular value decomposition. Let $\mathbf{X} = \widetilde{\mathbf{O}}_1\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{O}}_2^{\mathrm{T}}$ be the (usual) singular value decomposition of $\mathbf{X}$ and let

$$\mathbf{E} = \mathrm{diag}(1, \cdots, 1, -1) \qquad (31.19)$$

be the identity matrix with its last entry changed to -1. Let $\mathbf{O}_1 = \widetilde{\mathbf{O}}_1\mathbf{E}^{\delta_1}$ where $\delta_1 = 0$ if $\widetilde{\mathbf{O}}_1 \in \mathcal{SO}(p)$ and $\delta_1 = 1$ otherwise.

Similarly define δ_2 and $\mathbf{O}_2$. Finally write $\mathbf{\Lambda} = \widetilde{\mathbf{\Lambda}}\mathbf{E}^{\delta_1+\delta_2}$. Then (31.16) is valid with $\mathbf{O}_1, \mathbf{O}_2 \in \mathcal{SO}(p)$ and $\lambda_1 \ge \cdots \ge \lambda_{p-1} \ge |\lambda_p|$.

This is the modified singular value decomposition.

The least squares estimates, subject to the contraint $\widehat{\mathbf{A}} \in \mathcal{SO}(p)$, is still given by (31.18) when a modified singular value decomposition is used for $\mathbf{X}$.

31.2.4 Example: Least Squares Estimates for the Hands Data

Consider, for example, the hands data in Table 31.1. For this data

$$\begin{aligned}
\overline{\boldsymbol{u}} &= \begin{pmatrix} 7.8975 \\ 12.7542 \\ 14.3067 \end{pmatrix}\,, \quad \overline{\boldsymbol{v}} = \begin{pmatrix} 9.2500 \\ 11.3633 \\ 15.0950 \end{pmatrix}\,, \\
\mathbf{X} &= \left[\begin{pmatrix} 5.17 \\ 11.30 \\ 16.18 \end{pmatrix} - \overline{\boldsymbol{u}}\right]\left[\begin{pmatrix} 5.91 \\ 11.16 \\ 16.55 \end{pmatrix} - \overline{\boldsymbol{v}}\right]^{\mathrm{T}} + \\
&\quad \cdots + \left[\begin{pmatrix} 8.68 \\ 12.71 \\ 13.67 \end{pmatrix} - \overline{\boldsymbol{u}}\right]\left[\begin{pmatrix} 10.15 \\ 12.17 \\ 14.44 \end{pmatrix} - \overline{\boldsymbol{v}}\right]^{\mathrm{T}} \\
&= \begin{pmatrix} 34.0963 & -6.9083 & 3.5769 \\ 17.3778 & -4.9028 & -5.6605 \\ -2.3940 & -5.7387 & 57.8598 \end{pmatrix}\,.
\end{aligned}$$

The singular value decomposition $\mathbf{X} = \mathbf{O}_1\mathbf{\Lambda}\mathbf{O}_2^{\mathrm{T}}$ is given by

$$\mathbf{O}_1 = \begin{pmatrix} 0.0465 & -0.8896 & -0.4544 \\ -0.1012 & -0.4567 & 0.8838 \\ 0.9938 & -0.0048 & 0.1112 \end{pmatrix}\,,$$

$$\mathbf{O}_2 = \begin{pmatrix} -0.0436 & -0.9764 & -0.2114 \\ -0.0944 & 0.2147 & -0.9721 \\ 0.9946 & -0.0224 & -0.1015 \end{pmatrix}$$

$$\mathbf{\Lambda} = \mathrm{diag}(58.5564, 39.1810, 1.8855)\,.$$

Hence (31.18) yields

$$\widehat{\mathbf{A}} = \begin{pmatrix} 0.9627 & 0.2635 & -0.0621 \\ 0.2463 & -0.9477 & -0.2030 \\ 0.1123 & -0.1801 & 0.9772 \end{pmatrix}\,,$$

$$\widehat{\gamma} = 0.9925\,, \qquad (31.20)$$

$$\widehat{\boldsymbol{b}} = \begin{pmatrix} -0.7488 & 24.3115 & 2.6196 \end{pmatrix}^{\mathrm{T}}\,.$$

Notice that $\det(\widehat{\mathbf{A}}) = -1$ so $\widehat{\mathbf{A}} \notin \mathcal{SO}(3)$. We expect this result since, as previously remarked, the left and right hands have different orientations. The value of $\widehat{\gamma}$ is somewhat puzzling since the subject is right-handed and one would expect, therefore, $\gamma > 1$. Although, as we will see in Sect. 31.4, the difference between $\widehat{\gamma}$ and 1 is not significant, a better estimate would have been achieved if the L_1 objective function (31.11) were numerically minimized instead. In this case $\widehat{\gamma} = 1.0086$. Our analysis will show that the hands data set has an outlier and we see here an example of the superior resistance of L_1 estimates to outliers.

31.2.5 Least Squares Estimation in the Spherical Regression Model

Least squares estimation for the spherical regression model is similar to least squares estimation in the Procrustes model. Let $\mathbf{X} = \sum_i \boldsymbol{u}_i\boldsymbol{v}_i^{\mathrm{T}}$ and define $\mathbf{O}_1, \mathbf{O}_2 \in \mathcal{O}(p)$ using a singular value decomposition of $\mathbf{X}$. Then $\widehat{\mathbf{A}} = \mathbf{O}_2\mathbf{O}_1^{\mathrm{T}}$. If, on the other hand, it is desired to constrain $\widehat{\mathbf{A}}$ to $\mathcal{SO}(p)$, one defines $\mathbf{O}_1, \mathbf{O}_2 \in \mathcal{SO}(p)$ using a modified singular value decomposition and, again, $\widehat{\mathbf{A}} = \mathbf{O}_2\mathbf{O}_1^{\mathrm{T}}$.

31.3 Geometry of $\mathcal{O}(p)$ and $\mathcal{SO}(p)$

$\mathcal{O}(p)$ and $\mathcal{SO}(p)$ arise because they give distance-preserving transformations of $\mathbb{R}^p$, and to formulate properly the statistical properties of $\widehat{\mathbf{A}}$ defined by (31.18), it is important to understand the geometry of these two groups.

31.3.1 $\mathcal{SO}(p)$ for $p = 2$

For $p = 2$,

$$\mathcal{SO}(2) = \left\{ \Phi_2(h) = \begin{pmatrix} \cos(h) & -\sin(h) \\ \sin(h) & \cos(h) \end{pmatrix} \middle| \, h \in \mathbb{R}^1 \right\} . \tag{31.21}$$

Physically $\Phi_2(h)$ represents a rotation of $\mathbb{R}^2$ by an angle of h radians. Since $\Phi_2(h) = \Phi_2(h + 2\pi)$, $\mathcal{SO}(2)$ is geometrically a circle.

Since each element of $\mathcal{SO}(2)$ has four entries, it is tempting to think of $\mathcal{SO}(2)$ as four-dimensional. However as (31.21) makes clear, $\mathcal{SO}(2)$ can be described by one parameter $h \in \mathbb{R}^1$. Thus $\mathcal{SO}(2)$ is really one-dimensional. Suppose we were constrained to live on a circle Ω_2 (instead of the sphere Ω_3). At each point on Ω_2 we can only travel to our left or to our right, and, if our travels were limited, it would appear as if we only had one-dimensional travel. Mathematicians describe this situation by saying that $\mathcal{SO}(2)$ is a one-dimensional *manifold*.

Notice also $\Phi_2(0) = \mathbf{I}_2$ and that, if h is small, then $\Phi_2(h)$ is close to $\mathbf{I}_2$. Thus, if h is small, $\Phi_2(h)\boldsymbol{x}$ is close to $\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^2$. As we shall see, this simple observation is key to understanding our approach to the statistical properties of $\widehat{\mathbf{A}}$.

31.3.2 $\mathcal{SO}(p)$ for $p = 3$

$\mathcal{SO}(3)$ can be described as the collection of all rotations in $\mathbb{R}^3$. That is

$$\mathcal{SO}(3) = \left\{ \Phi_3(\boldsymbol{h}) \mid \boldsymbol{h} \in \mathbb{R}^3 \right\} , \tag{31.22}$$

where $\Phi_3(\boldsymbol{h})$ is right-hand rule rotation of $||\boldsymbol{h}||$ radians around the axis $||\boldsymbol{h}||^{-1}\boldsymbol{h}$. Writing $\theta = ||\boldsymbol{h}||$ and $\xi = ||\boldsymbol{h}||^{-1}\boldsymbol{h}$, so that ξ is a unit-length three-dimensional vector, it can be shown that

$$\begin{aligned} &\Phi_3(\boldsymbol{h}) \\ &= \Phi_3(\theta\xi) \\ &= \cos(\theta)\mathbf{I}_3 + \sin(\theta)M_3(\xi) + [1 - \cos(\theta)]\,\xi\xi^{\mathrm{T}} , \end{aligned} \tag{31.23}$$

where

$$M_3(\xi) = M_3 \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix} .$$

Thus, although each $\mathbf{A} \in \mathcal{SO}(3)$ has nine entries, $\mathcal{SO}(3)$ is actually a three-dimensional manifold.

Again we notice that $\Phi_3(\mathbf{0}) = \mathbf{I}_3$ and that if $||\boldsymbol{h}||$ is small then $\Phi_3(\boldsymbol{h})\boldsymbol{x}$ is close to $\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^3$.

For future use, we note that if $\mathbf{C} \in \mathcal{SO}(3)$, then the axis ξ of the rotation represented by $\mathbf{C}$ satisfies $\mathbf{C}\xi = \xi$. Thus ξ is the eigenvector associated to the eigenvalue 1 of $\mathbf{C}$. By re-representing $\mathbf{C}$ in an orthonormal basis which includes ξ, one can show that the angle of rotation θ of the rotation represented by $\mathbf{C}$ satisfies $1 + 2\cos(\theta) = \mathrm{Tr}(\mathbf{C})$. Thus, if ξ and θ are calculated in this way, $\Phi_3(\theta\xi) = \mathbf{C}$.

31.3.3 $\mathcal{SO}(p)$ and $\mathcal{O}(p)$, for General p, and the Matrix Exponential Map

For general p, let $\mathbf{H}$ be a $p \times p$ *skew-symmetric* matrix; that is

$$\mathbf{H}^{\mathrm{T}} = -\mathbf{H} .$$

We define the matrix exponential map by

$$\exp(\mathbf{H}) = \sum_{k=0}^{k=\infty} \frac{\mathbf{H}^k}{k!} .$$

It can be shown that the skew-symmetry condition implies that $\exp(\mathbf{H})\left[\exp(\mathbf{H})\right]^{\mathrm{T}} = \mathbf{I}_p$ and indeed

$$\mathcal{SO}(p) = \left[\exp(\mathbf{H}) \mid \mathbf{H} \text{ is skew-symmetric}\right] . \tag{31.24}$$

A skew-symmetric matrix must have zeros on its main diagonal and its entries below the main diagonal are determined by its entries above the main diagonal. Thus the skew-symmetric $p \times p$ matrices have $p(p-1)/2$ independent entries and hence $\mathcal{SO}(p)$ is a manifold with dimension $p(p-1)/2$.

Let $\mathbf{0}_{(p,p)}$ be a $p \times p$ matrix of zeros. Then

$$\exp(\mathbf{0}_{(p,p)}) = \mathbf{I}_p . \tag{31.25}$$

Thus, if the entries of $\mathbf{H}$ are small (in absolute value), then $\exp(\mathbf{H})$ will be close to the identity matrix.

For $p = 3$, it can be shown, by using (31.23), that $\Phi_3(\boldsymbol{h}) = \exp\left[M_3(\boldsymbol{h})\right]$ for $\boldsymbol{h} \in \mathbb{R}^3$. Similarly we define for

$h \in \mathbb{R}^1$ the skew-symmetric matrix

$$M_2(h) = \begin{pmatrix} 0 & -h \\ h & 0 \end{pmatrix}$$

and it follows that $\Phi_2(h) = \exp[M_2(h)]$. Thus (31.21) and (31.22) are indeed special cases of (31.24).

$\mathcal{O}(p)$ has two connected components; one is $\mathcal{SO}(p)$ and the other is

$$\mathcal{SO}(p)\boldsymbol{E} = \{\mathbf{A}\boldsymbol{E} \mid \mathbf{A} \in \mathcal{SO}(p)\}\,,$$

where $\boldsymbol{E}$ has been previously defined in (31.19). Notice that $\boldsymbol{E}$ is a reflection of $\mathbb{R}^p$ through the $(p-1)$-dimensional hyperplane perpendicular to the last coordinate vector. Indeed all reflections of $\mathbb{R}^p$ are in $\mathcal{O}(p)$.

31.3.4 Geometry and the Distribution of *M*-Estimates

So, heuristically speaking, suppose we have estimates $\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}}\right)$ which minimize an objective function of the form (31.13). What values of the unknown parameters $(\mathbf{A}, \gamma, \boldsymbol{b})$ should we consider as reasonable given the data? The obvious answer, which is fully consistent with the usual practices of statistics, is those $(\mathbf{A}, \gamma, \boldsymbol{b})$ which do not excessively degrade the fit of the best-fit parameters $\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}}\right)$; that is those $(\mathbf{A}, \gamma, \boldsymbol{b})$ for which

$$\begin{aligned}\rho(\mathbf{A}, \gamma, \boldsymbol{b}) - \rho\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}}\right) \\ = \sum_i \left[\rho_0(||\boldsymbol{v}_i - \gamma\mathbf{A}\boldsymbol{u}_i - \boldsymbol{b}||)\right. \\ \left. - \rho_0(||\boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}\boldsymbol{u}_i - \widehat{\boldsymbol{b}}||)\right]\end{aligned}$$

is not too large.

Recall that, for $p = 3$, if $\boldsymbol{h}$ is small, then $\Phi_3(\boldsymbol{h})\boldsymbol{u}_i$ will be close to $\boldsymbol{u}_i$. This suggests writing

$$\widehat{\mathbf{A}} = \mathbf{A}\Phi_3(\widehat{\boldsymbol{h}})\,, \tag{31.26}$$

where $\widehat{\boldsymbol{h}} \in \mathbb{R}^3$. Then $\mathbf{A}\boldsymbol{u}_i = \widehat{\mathbf{A}}\Phi_3(-\widehat{\boldsymbol{h}})\boldsymbol{u}_i$ will be close to $\widehat{\mathbf{A}}\boldsymbol{u}_i$ when $\widehat{\boldsymbol{h}}$ is small. Rather than focus on the distribution of $\widehat{\mathbf{A}}$, we will focus on the distribution of the *deviation* of $\widehat{\mathbf{A}}$ from $\mathbf{A}$ as measured by the (hopefully) small vector $\widehat{\boldsymbol{h}}$.

Similarly, for $p = 2$, we will write

$$\widehat{\mathbf{A}} = \mathbf{A}\Phi_2(\widehat{h})\,, \tag{31.27}$$

where $\widehat{h} \in \mathbb{R}^1$. For general p, one writes

$$\widehat{\mathbf{A}} = \mathbf{A}\exp(\widehat{\mathbf{H}})\,, \tag{31.28}$$

where $\widehat{\mathbf{H}}$ is $p \times p$ skew-symmetric.

The most elementary procedures in statistics are based upon the fact

> *If $X_1, \cdots, X_n$ are independent and each X_i is distributed $N(\mu, \sigma^2)$, then $\overline{X}$ is distributed $N(\mu, \sigma^2/n)$.*

An equivalent result is

> *If $X_1, \cdots, X_n$ are independent and each X_i is distributed $N(\mu, \sigma^2)$, then $\overline{X} - \mu$ is distributed $N(0, \sigma^2/n)$.*

In the latter form, we have an estimator (in this case $\overline{X}$) and the distribution of the *deviation* $\widehat{h} = \overline{X} - \mu$ of the estimator from the unknown parameter μ. This is sufficient for both confidence intervals and hypothesis testing and is analogous to what we propose to do in Sect. 31.4.

We note that $\Phi_3(\boldsymbol{h}) = \mathbf{I}_3$ whenever $||\boldsymbol{h}|| = 2\pi$. This implies that Φ_3 will have a singularity as $||\boldsymbol{h}|| \to 2\pi$. However, Φ_3 behaves very well for small $\boldsymbol{h}$ and hence (31.26) is a good way to parameterize $\widehat{\mathbf{A}}$ close to $\mathbf{A}$.

All parameterizations of $\mathcal{SO}(3)$ have singularities somewhere. By using parameterizations such as (31.26), (31.27), or (31.28), we put those singularities far away from the region of interest, that is far away from $\mathbf{A}$. As we will see in Sects. 31.4 and 31.5, the result is very clean mathematics. However, some formulations of Euler angles [31.7] have a singularity at $\boldsymbol{h} = \mathbf{0}$. This means that Euler angles are an especially poor parameterization of small rotations in $\mathcal{SO}(3)$ (that is, for $\mathbf{A}$ close to $\mathbf{I}_3$) and that, if we were to repeat the calculations of Sect. 31.4 and 31.5 using Euler angles, the results would be much messier.

31.3.5 Numerical Calculation of *M*-Estimates for the Procrustes Model

We use here the geometric insights into $\mathcal{SO}(p)$ to propose a method of minimizing the objective function (31.13) for the Procrustes model. The simplifications necessary to minimize the objective function (31.14) for the spherical regression model should be reasonably clear.

In what follows, it will be convenient to rewrite the Procrustes model

$$\boldsymbol{v}_i = \gamma\mathbf{A}\boldsymbol{u}_i + \boldsymbol{b} + \text{error}$$

in the equivalent form

$$\boldsymbol{v}_i = \gamma\mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) + \boldsymbol{\beta} + \text{error}\,, \tag{31.29}$$

where $\boldsymbol{\beta} = \gamma\mathbf{A}\overline{\boldsymbol{u}} + \boldsymbol{b}$.

Let $\psi(s) = \rho_0'(s)$. Differentiating (31.13) with respect to γ and $\boldsymbol{\beta}$ we get that the M-estimates $(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{\beta}})$ must satisfy.

$$0 = \sum_i \psi(s_i) s_i^{-1} \left[\boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}} \right]^{\mathrm{T}} \times \widehat{\mathbf{A}}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \,, \tag{31.30}$$

$$0 = \sum_i \psi(s_i) s_i^{-1} \left[\boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}} \right] \,, \tag{31.31}$$

where $s_i = \left|\left| \boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}} \right|\right|$.

To differentiate (31.13) with respect to $\mathbf{A}$, we note that, if $\mathbf{H}$ is any skew-symmetric matrix, and using (31.25),

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0} \left\{ \sum_i \rho_0 \left[\left|\left| \boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}\exp(t\mathbf{H}) \right.\right.\right. \\ &\quad \left.\left.\left. \times (\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}} \right|\right| \right] \right\} \\ &= -\gamma \sum_i \psi(s_i) s_i^{-1} \left[\boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}\,(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}} \right]^{\mathrm{T}} \\ &\quad \times \widehat{\mathbf{A}}\mathbf{H}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \\ &= -\gamma \mathrm{Tr}(\widetilde{\mathbf{X}}\mathbf{H}), \end{aligned}$$

where

$$\widetilde{\mathbf{X}} = \sum_i \psi(s_i) s_i^{-1} \, (\boldsymbol{u}_i - \overline{\boldsymbol{u}}) \left[\boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}} \right]^{\mathrm{T}} \widehat{\mathbf{A}} \,.$$

Since $\mathbf{H}$ is any skew-symmetric matrix, $\widetilde{\mathbf{X}}$ is symmetric. Equivalently

$$\mathbf{X} = \sum_i \psi(s_i) s_i^{-1} (\boldsymbol{u}_i - \overline{\boldsymbol{u}})(\boldsymbol{v}_i - \widehat{\boldsymbol{\beta}})^{\mathrm{T}} \widehat{\mathbf{A}} \text{ is symmetric} \,. \tag{31.32}$$

Equations (31.32), (31.30), and (31.31) lead to the following iterative minimization algorithm. Start with the least squares solution given in Sect. 31.2.3 and use these estimates to calculate s_i. Using these s_i and the current guess for $\widehat{\mathbf{A}}$, solve (31.30) and (31.31) to update the guesses for $\widehat{\gamma}$ and $\widehat{\beta}$. Now writing $\mathbf{X} = \mathbf{O}_1 \mathbf{\Lambda} \mathbf{O}_2^{\mathrm{T}}$ for the singular value decomposition of $\mathbf{X}$, the next guess for $\widehat{\mathbf{A}}$ is $\mathbf{O}_2\mathbf{O}_1^{\mathrm{T}}$. This yields a minimum in $\mathcal{O}(p)$. If minimization in $\mathcal{SO}(p)$ is desired, a modified singular value decomposition is used for $\mathbf{X}$ instead. Having updated the guesses for $\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{\beta}}\right)$, we now iterate.

For example consider the hands data of Table 31.1. We calculate the L_1 estimate for which $\psi(s) = 1$. Starting with the least squares estimates in (31.20), we convert $\widehat{\boldsymbol{b}}$ to

$$\widehat{\beta} = \widehat{\gamma}\widehat{\mathbf{A}}\overline{\boldsymbol{u}} + \widehat{\boldsymbol{b}} = \begin{pmatrix} 9.2500 & 11.3633 & 15.0950 \end{pmatrix}^{\mathrm{T}} . \tag{31.33}$$

We use these least squares estimate as an initial guess; a single iteration of the minimization algorithm yields the updated guess

$$\begin{aligned} \widehat{\mathbf{A}} &= \begin{pmatrix} 0.9569 & 0.2823 & -0.0690 \\ 0.2614 & -0.9399 & -0.2199 \\ 0.1269 & -0.1924 & 0.9731 \end{pmatrix} , \\ \widehat{\gamma} &= 1.0015 \,, \\ \widehat{\beta} &= \begin{pmatrix} 9.2835 & 11.4092 & 15.0851 \end{pmatrix}^{\mathrm{T}} . \end{aligned}$$

Convergence is achieved after around a dozen iterations. We arrive at the L_1 estimates

$$\begin{aligned} \widehat{\mathbf{A}} &= \begin{pmatrix} 0.9418 & 0.3274 & -0.0760 \\ 0.3045 & -0.9268 & -0.2200 \\ 0.1425 & -0.1840 & 0.9725 \end{pmatrix} , \\ \widehat{\gamma} &= 1.0086 \,, \\ \widehat{\beta} &= \begin{pmatrix} 9.2850 & 11.4255 & 15.0883 \end{pmatrix}^{\mathrm{T}} . \end{aligned} \tag{31.34}$$

31.4 Statistical Properties of *M*-Estimates

31.4.1 The Σ Matrix and the Geometry of the u_i

Let $\mathbf{\Sigma}$ be the $p \times p$ matrix

$$\mathbf{\Sigma} = n^{-1} \sum_i (\boldsymbol{u}_i - \overline{\boldsymbol{u}})(\boldsymbol{u}_i - \overline{\boldsymbol{u}})^{\mathrm{T}}$$

$\mathbf{\Sigma}$ is nonnegative definite symmetric and hence its eigenvalues are real and its eigenvectors form an orthonormal basis of $\mathbb{R}^p$. We can use this eigen-decomposition of $\mathbf{\Sigma}$ to summarize the geometry of the point $\boldsymbol{u}_i$. More specifically, let $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ be the eigenvalues of $\mathbf{\Sigma}$ with corresponding eigenvectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_p$. Then $\boldsymbol{e}_1$ points in the direction of the greatest variation in the $\boldsymbol{u}_i$, and $\boldsymbol{e}_p$ in the direction of the least variation.

31.4.2 Example: Σ for the Hands Data

For example, for the data of Table 31.1,

$$\overline{\boldsymbol{u}} = \begin{pmatrix} 7.8975 \\ 12.7542 \\ 14.3067 \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \frac{1}{12}\left\{\left[\begin{pmatrix} 5.17 \\ 11.30 \\ 16.18 \end{pmatrix} - \overline{\boldsymbol{u}}\right]\left[\begin{pmatrix} 5.17 \\ 11.30 \\ 16.18 \end{pmatrix} - \overline{\boldsymbol{u}}\right]^{\mathrm{T}} + \cdots + \left[\begin{pmatrix} 8.68 \\ 12.71 \\ 13.67 \end{pmatrix} - \overline{\boldsymbol{u}}\right]\left[\begin{pmatrix} 8.68 \\ 12.71 \\ 13.67 \end{pmatrix} - \overline{\boldsymbol{u}}\right]^{\mathrm{T}}\right\}$$

$$= \begin{pmatrix} 2.6249 & 1.2525 & 0.1424 \\ 1.2525 & 0.8095 & -0.5552 \\ 0.1424 & -0.5552 & 4.9306 \end{pmatrix},$$

$$\lambda_1 = 5.004\,, \quad \lambda_2 = 3.255\,, \quad \lambda_3 = 0.1054\,,$$

$$\boldsymbol{e}_1 = \begin{pmatrix} -0.0115 \\ -0.1346 \\ 0.9908 \end{pmatrix}, \quad \boldsymbol{e}_2 = \begin{pmatrix} -0.8942 \\ -0.4420 \\ -0.0704 \end{pmatrix},$$

$$\boldsymbol{e}_3 = \begin{pmatrix} -0.4474 \\ 0.8869 \\ 0.1152 \end{pmatrix}.$$

Examining the data of Table 31.1, one sees that $\overline{\boldsymbol{u}}$ is close to point G, the center of the left palm. Examining the displacement of G to C, top of the middle finger, it is evident that left hand was close to vertically oriented. This is the direction $\boldsymbol{e}_1$. Examining the displacement of G to E, the top of the thumb, it appears that the left thumb was pointed in roughly the direction of the x-axis. This is the direction of $-\boldsymbol{e}_2$. Thus the left hand was roughly parallel to the x–z plane. The normal vector to the plane of the left hand is thus approximately parallel to the y-axis. This is the direction of $\boldsymbol{e}_3$. Notice that λ_3 is much smaller than λ_1 or λ_2, indicating that the thickness of the hand is much smaller than its length or breadth.

31.4.3 Statistical Assumptions for the Procrustes Model

Before giving the statistical properties of $\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}}\right)$ it is necessary to make explicit the statistical assumptions of the Procrustes model (31.2). These assumptions are:

- $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n \in \mathbb{R}^p$ are fixed (non-random) vectors.
- $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^p$ are independent random vectors.
- The distribution of $\boldsymbol{v}_i$ is of the form $f_0(s_i)$, where $s_i = ||\boldsymbol{v}_i - \gamma \mathbf{A}\boldsymbol{u}_i - \boldsymbol{b}||$. Here $(\mathbf{A}, \gamma, \boldsymbol{b})$ are unknown, $\mathbf{A} \in \mathcal{SO}(p)$ or $\mathcal{O}(p)$, γ is a positive real constant, and $\boldsymbol{b} \in \mathbb{R}^p$.

The most obvious example of a suitable distribution f_0 is

$$f_0(s) = (2\pi\sigma^2)^{-p/2}\, \mathrm{e}^{-\frac{s^2}{2\sigma^2}} \tag{31.35}$$

for a fixed constant σ^2. In what follows, we will not need to know the value of σ^2. In fact, we will not even need to know the form of f_0, only that the distribution of $\boldsymbol{v}_i$ depends only upon its distance s_i from $\gamma\mathbf{A}\boldsymbol{u}_i + \boldsymbol{b}$.

The distribution (31.35) is a multivariate normal distribution with mean vector $\gamma\mathbf{A}\boldsymbol{u}_i + \boldsymbol{b}$ and covariance matrix $\sigma^2\mathbf{I}_p$. Equivalently, the p components of $\boldsymbol{v}_i$ are independent and each has variance σ^2. If the components of $\boldsymbol{v}_i$ were to have different variances, then the distribution of $\boldsymbol{v}_i$ would not satisfy the Procrustes model assumptions.

In essence we assume that $\boldsymbol{v}_i$ is isotropically (i.e., that all directions are the same) distributed around its mean vector.

31.4.4 Theorem (Distribution of $(\widehat{\mathbf{A}},\widehat{\gamma},\widehat{\boldsymbol{b}})$ for the Procrustes Model)

Suppose $\left(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}}\right)$ minimize an objective function of the form (31.13). Let $\boldsymbol{\beta} = \gamma\mathbf{A}\overline{\boldsymbol{u}} + \boldsymbol{b}$ and $\widehat{\boldsymbol{\beta}} = \widehat{\gamma}\widehat{\mathbf{A}}\overline{\boldsymbol{u}} + \widehat{\boldsymbol{b}}$. Then

- $\widehat{\mathbf{A}}$, $\widehat{\gamma}$, and $\widehat{\boldsymbol{\beta}}$ are independent;
- $\widehat{\boldsymbol{\beta}}$ is distributed multivariate normal with mean $\boldsymbol{\beta}$ and covariance matrix $\frac{k}{n}\mathbf{I}_p$;
- If $p = 2$, write $\widehat{\mathbf{A}} = \mathbf{A}\Psi_2(\widehat{h})$, for $\widehat{h} \in \mathbb{R}^1$. Then $\widehat{h}$ is normally distributed with mean 0 and variance $\frac{k}{n\mathrm{Tr}(\Sigma)}$;
- If $p = 3$, write $\widehat{\mathbf{A}} = \mathbf{A}\Psi_3(\widehat{\boldsymbol{h}})$, for $\widehat{\boldsymbol{h}} \in \mathbb{R}^3$. Let $\Sigma = \lambda_1\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} + \lambda_2\boldsymbol{e}_2\boldsymbol{e}_2^{\mathrm{T}} + \lambda_3\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}}$ be the spectral decomposition of $\boldsymbol{\Sigma}$. Then $\widehat{\boldsymbol{h}}$ is distributed trivariate normal with mean $\mathbf{0}$ and covariance matrix

$$\frac{k}{n}\Big[(\lambda_2+\lambda_3)^{-1}\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} + (\lambda_3+\lambda_1)^{-1}\boldsymbol{e}_2\boldsymbol{e}_2^{\mathrm{T}} + (\lambda_1+\lambda_2)^{-1}\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}}\Big].$$

- For general p, write $\widehat{\mathbf{A}} = \mathbf{A}\exp(\widehat{\mathbf{H}})$, where $\widehat{\mathbf{H}}$ is $p \times p$ skew-symmetric. Then $\widehat{\mathbf{H}}$ has a multivariate normal density proportional to $\exp\left[-\frac{n}{2k}\mathrm{Tr}(\widehat{\mathbf{H}}^{\mathrm{T}}\boldsymbol{\Sigma}\widehat{\mathbf{H}})\right]$;
- $\widehat{\gamma}$ is normally distributed with mean γ and variance $\frac{k}{n\mathrm{Tr}(\boldsymbol{\Sigma})}$.

These results are asymptotic, that is they are large-sample approximate distributions.

The constant k is defined to be

$$k = \frac{pE\left[\psi(s)^2\right]}{E^2\left[\psi'(s)+(p-1)\psi(s)s^{-1}\right]}, \tag{31.36}$$

where $\psi(s) = \rho_0'(s)$. Thus k can be estimated from the sample by

$$\widehat{k} = \frac{np\sum_i \psi(s_i)^2}{\left\{\sum_i\left[\psi'(s_i)+(p-1)\psi(s_i)s_i^{-1}\right]\right\}^2}, \tag{31.37}$$

where $s_i = \left|\left|\boldsymbol{v}_i - \widehat{\gamma}\widehat{\mathbf{A}}\boldsymbol{u}_i - \widehat{\boldsymbol{b}}\right|\right|$.

Theorem 31.4.4 is proven in *Chang* and *Ko* [31.3]. (In [31.3], s is defined to be $s = ||\boldsymbol{v} - \widehat{\gamma}\widehat{\mathbf{A}}\boldsymbol{u} - \widehat{\boldsymbol{b}}||^2$ and this causes the formulas (31.36) and (31.37) to be somewhat different there.)

31.4.5 Example: A Test of $\gamma = 1$

For the hands data, the least squares estimates were given in Example 31.2.4. Table 31.2 gives the calculation of the s_i. Substituting $p = 3$, $\rho_0(s) = s^2$, $\psi(s) = 2s$ into (31.37), $\widehat{k} = (3n)^{-1}\sum_i s_i^2 = 0.0860$.

To test if the two hands are the same size, we test $\gamma = 1$. Using Example 31.4.2, $\mathrm{Tr}(\boldsymbol{\Sigma}) = 8.365$. Hence the variance of $\widehat{\gamma}$ is 0.000 860 and hence its standard error is 0.0293. Since $\widehat{\gamma} = 0.9925$, we see that $\widehat{\gamma}$ is not significantly different from 1.

The L_1 estimate of γ is 1.0086. To calculate the standard error of this estimate, we use $\rho_0(s) = s$ and $\psi(s) = 1$. Hence for the L_1 estimate, (31.37) yields $\widehat{k} = 0.75\left(n^{-1}\sum_i s_i^{-1}\right)^{-2}$. After recomputing the s_i using L_1 estimates of $(\mathbf{A}, \gamma, \boldsymbol{b})$, we obtain $\widehat{k} = 0.023$. Thus the L_1 estimate of γ has a standard error of 0.0150 and this estimate is also not significantly different from 1.

Apparently, the two hands have the same size.

General statistical theory implies that if the $\boldsymbol{v}_i$ were really normally distributed, the least squares estimates would be the most efficient. In other words, least squares estimates should have the smallest standard errors. Evidently this is not true for the hands data and it appears that this data is not, in fact, normally distributed.

31.4.6 Example: A Test on A

As discussed in 31.4.2, the eigenvector $\boldsymbol{e}_3$ of $\boldsymbol{\Sigma}$ is perpendicular to the plane of the left palm. It might be of interest to test if the two hands have the same orientation; that is, after reflecting the left hand in the plane perpendicular to $\boldsymbol{e}_3$, do the fingers and thumb of the two hands point in the same directions. We formulation this hypothesis as $H_0 : \mathbf{A} = \mathbf{R}_{\boldsymbol{e}_3}$ where $\mathbf{R}_{\boldsymbol{e}_3}$ is the matrix of the reflection in plane perpendicular to $\boldsymbol{e}_3$.

$$\begin{aligned}\mathbf{R}_{\boldsymbol{e}_3} &= \mathbf{I}_3 - 2\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &\quad -2\begin{pmatrix} -0.4474 \\ 0.8869 \\ 0.1152 \end{pmatrix}\begin{pmatrix} -0.4474 \\ 0.8869 \\ 0.1152 \end{pmatrix}^{\mathrm{T}} \\ &= \begin{pmatrix} 0.5996 & 0.7936 & 0.1031 \\ 0.7936 & -0.5731 & -0.2044 \\ 0.1031 & -0.2044 & 0.9734 \end{pmatrix},\end{aligned}$$

$\widehat{\boldsymbol{h}}$ is defined by

$$\begin{aligned}\Phi_3(\widehat{\boldsymbol{h}}) &= \mathbf{R}_{\boldsymbol{e}_3}^{\mathrm{T}}\widehat{\mathbf{A}} \\ &= \begin{pmatrix} 0.7843 & -0.6127 & -0.0976 \\ 0.5999 & 0.7890 & -0.1327 \\ 0.1583 & 0.0455 & 0.9863 \end{pmatrix},\end{aligned} \tag{31.38}$$

where $\widehat{\mathbf{A}}$ was calculated in 31.2.4.

To solve for $\widehat{\boldsymbol{h}}$ we use the results at the end of Sect. 31.3.2. The matrix of (31.38) has an eigenvector of $\xi = (0.1395 \quad -0.2003 \quad 0.9494)^{\mathrm{T}}$ corresponding to the eigenvalue of 1. Its angle of rotation is given by

$$\theta = \arccos\left[0.5\mathrm{Tr}\left(\mathbf{R}_{\boldsymbol{e}_3}^{\mathrm{T}}\widehat{\mathbf{A}}\right) - 0.5\right] = 0.6764\,.$$

Thus $\widehat{\boldsymbol{h}} = \theta\boldsymbol{\xi} = (0.0944 \quad -0.1355 \quad 0.6422)^{\mathrm{T}}$.

By Theorem 31.4.4, if H_0 is true, $\widehat{\boldsymbol{h}}$ is trivariate normally distributed with mean $\mathbf{0}$ and covariance matrix

$$\begin{aligned}&\frac{k}{n}\Big[(\lambda_2+\lambda_3)^{-1}\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} + (\lambda_3+\lambda_1)^{-1}\boldsymbol{e}_2\boldsymbol{e}_2^{\mathrm{T}} \\ &+(\lambda_1+\lambda_2)^{-1}\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}}\Big].\end{aligned}$$

The constant k was estimated in 31.4.5 and the λ_i and $\boldsymbol{e}_i$ were calculated in 31.4.2. Using these calculations, the covariance matrix of $\widehat{\boldsymbol{h}}$ is estimated to be

$$\begin{aligned}&\widehat{\mathrm{Cov}}(\widehat{\boldsymbol{h}}) \\ &= \begin{pmatrix} 0.001\,296 & 0.000\,2134 & 0.000\,019\,23 \\ 0.000\,2134 & 0.000\,9951 & -0.000\,1520 \\ 0.000\,019\,23 & -0.000\,1520 & 0.002\,112 \end{pmatrix}.\end{aligned}$$

Table 31.2 Calculation of residual lengths for data from Table 31.1

	Predicted $\widehat{v}_i$ $\widehat{\gamma}\widehat{A}u_i+\widehat{b}$			Residual $v_i-\widehat{v}_i$			s_i $\|\|v_i-\widehat{v}_i\|\|$
A	6.148	11.687	16.868	−0.238	−0.527	−0.318	0.660
B	8.475	10.969	18.207	0.155	−0.349	0.123	0.401
C	9.620	10.962	18.654	0.470	−0.362	−0.014	0.593
D	11.080	10.457	17.769	−0.190	0.493	0.131	0.544
E	13.206	10.816	13.865	−0.236	−0.686	0.015	0.726
F	8.691	11.176	13.247	0.099	0.034	−0.077	0.129
G	10.544	10.938	13.355	0.156	0.162	0.065	0.234
H	8.501	11.594	11.046	−0.031	−0.504	0.304	0.589
I	7.449	12.225	14.178	−0.169	0.295	−0.138	0.367
J	8.062	11.908	14.649	−0.012	0.512	−0.089	0.520
K	9.198	11.904	14.730	−0.128	0.486	0.130	0.519
L	10.026	11.724	14.573	0.125	0.446	−0.133	0.481

Under the null hypothesis

$$\chi^2 = \widehat{\boldsymbol{h}}^{\mathrm{T}}\widehat{\mathrm{Cov}}(\widehat{\boldsymbol{h}})^{-1}\widehat{\boldsymbol{h}} = 213$$

has an approximate χ^2 distribution with three degrees of freedom.

We emphatically conclude that, after reflecting the left hand, the orientations of the two hands are not the same.

31.4.7 Asymptotic Relative Efficiency of Least Squares and L_1 Estimates

Examining Theorem 31.4.4, we see that the covariance of the *M*-estimate $(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}})$ is determined, up to a constant k, by the geometry of the $\boldsymbol{u}_i$, as summarized by the matrix $\boldsymbol{\Sigma}$. Only the constant k, see (31.36), depends upon the probability distribution of the $\boldsymbol{v}_i$ and the objective function (31.13) that $(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}})$ minimize. Furthermore, a sample estimate of k, see (31.37) is available which does not require knowledge of the distribution of the $\boldsymbol{v}_i$.

Let $k(f_0, L_2)$ denote the constant k as defined in (31.36) when the underlying density is of the form f_0 and least squares (L_2) estimation is used, and $k(f_0, L_1)$ the corresponding value when L_1 estimation is used. The ratio $\mathrm{ARE}(L_1, L_2; f_0) = k(f_0, L_2)/k(f_0, L_1)$ is called the *asymptotic relative efficiency* of the L_1 to the least squares estimators at the density f_0.

We see that

$$\mathrm{ARE}(L_1, L_2; f_0) = \frac{\text{variance of least squares estimator}}{\text{variance of } L_1 \text{estimator}}, \tag{31.39}$$

where we recognize that both variances are matrices, but the two variance matrices are multiples of each other.

If f_0 is a p-dimensional normal density (31.35), it can be shown from (31.36) that

$$\mathrm{ARE}(L_1, L_2; N_p) = \frac{2\Gamma^2[(p+1)/2]}{p\Gamma^2(p/2)}. \tag{31.40}$$

We have used N_p in (31.40) to denote the p-dimensional normal density function.

The Γ function in (31.40) has the properties

$$\Gamma(1) = 1 \quad \Gamma(0.5) = \sqrt{\pi}$$
$$\Gamma(q+1) = q\Gamma(q).$$

Thus when $p = 2.3$

$$\mathrm{ARE}(L_1, L_2; N_2) = \frac{\pi}{4} = 0.785,$$
$$\mathrm{ARE}(L_1, L_2; N_3) = \frac{8}{3\pi} = 0.849. \tag{31.41}$$

$\mathrm{ARE}(L_1, L_2; N_p)$ increases to 1 as $p \to \infty$.

When the underlying distribution is normal, statistical theory indicates that least squares procedures are optimal, that is, they have the smallest variance. Using (31.39) and (31.41), we see that, even when the data is normal, the use of L_1 methods results in only an 8% penalty in standard error. And L_1 methods offer superior resistance to outliers.

Indeed, as we saw in Example 31.4.5, the standard error of the L_1 estimator was *smaller* than the standard error of the least squares estimator. Evidently the hands data set is long-tailed, that is it has more outliers than would be expected with normal data.

31.4.8 The Geometry of the Landmarks and the Errors in $\widehat{\mathbf{A}}$

In this section we will constrain our discussion to the case $p = 3$.

Suppose we write the estimate $\widehat{\mathbf{A}}$ in the form

$$\widehat{\mathbf{A}} = \mathbf{A}\Phi_3(\widehat{\boldsymbol{h}}) \,. \tag{31.42}$$

$\Phi_3(\widehat{\boldsymbol{h}})$ is a (hopefully) small rotation which expresses the deviation of the estimate $\widehat{\mathbf{A}}$ from the true value $\mathbf{A}$.

Recall that $\Phi_3(\widehat{\boldsymbol{h}})$ is a rotation of $||\widehat{\boldsymbol{h}}||$ radians around the axis $||\widehat{\boldsymbol{h}}||^{-1}\widehat{\boldsymbol{h}}$.

In particular $\Phi_3(\widehat{\boldsymbol{h}})^{-1} = \Phi_3(-\widehat{\boldsymbol{h}})$ and

$$\mathbf{A} = \widehat{\mathbf{A}}\Phi_3(-\widehat{\boldsymbol{h}}) \,.$$

According to Theorem 31.4.4, the covariance matrix of $\widehat{\boldsymbol{h}}$ has the form

$$\begin{aligned}\mathrm{Cov}(\widehat{\boldsymbol{h}}) = \frac{k}{n}\Big[&(\lambda_2+\lambda_3)^{-1}\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} + (\lambda_3+\lambda_1)^{-1}\boldsymbol{e}_2\boldsymbol{e}_2^{\mathrm{T}} \\ &+ (\lambda_1+\lambda_2)^{-1}\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}}\Big] \,,\end{aligned} \tag{31.43}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the eigenvalues of Σ with corresponding eigenvectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$. Since $\widehat{\boldsymbol{h}}$ is normally distributed

$$\chi^2 = \widehat{\boldsymbol{h}}^{\mathrm{T}}\left[\mathrm{Cov}(\widehat{\boldsymbol{h}})\right]^{-1}\widehat{\boldsymbol{h}}$$

is distributed χ^2 with three degrees of freedom.

Thus a confidence region for $\mathbf{A}$ is of the form

$$\left\{\widehat{\mathbf{A}}\Phi_3(-\widehat{\boldsymbol{h}}) \,\middle|\, \widehat{\boldsymbol{h}}^{\mathrm{T}}\left[\mathrm{Cov}(\widehat{\boldsymbol{h}})\right]^{-1}\widehat{\boldsymbol{h}} < \chi^2_{3,\alpha}\right\} \,, \tag{31.44}$$

where $\chi^2_{3,\alpha}$ is the appropriate critical point of a χ^2_3 distribution.

Let $\theta = ||\widehat{\boldsymbol{h}}||$ and $\xi = -||\widehat{\boldsymbol{h}}||^{-1}\widehat{\boldsymbol{h}}$ so that $\widehat{\boldsymbol{h}} = -\theta\xi$.

Thus $\Phi_3(-\widehat{\boldsymbol{h}})$ is a rotation of θ radians around the axis ξ.

Substituting (31.43) into the confidence region (31.44), we can re-express this confidence region as

$$\begin{aligned}\Big\{\widehat{\mathbf{A}}\Phi_3(\theta\xi) \mid \theta^2\frac{n}{k}\Big[&(\lambda_2+\lambda_3)(\xi^{\mathrm{T}}\boldsymbol{e}_1)^2 + (\lambda_3+\lambda_1) \\ &\times(\xi^{\mathrm{T}}\boldsymbol{e}_2)^2 + (\lambda_1+\lambda_2)(\xi^{\mathrm{T}}\boldsymbol{e}_3)^2\Big] < \chi^2_{3,\alpha}\Big\} \,.\end{aligned} \tag{31.45}$$

Now

$$\lambda_2+\lambda_3 \leq \lambda_3+\lambda_1 \leq \lambda_1+\lambda_2 \,.$$

Thus the confidence region (31.45) constrains θ the most (that is the limits on θ are the smallest) when ξ points in the direction $\boldsymbol{e}_3$. It bounds θ the least when ξ points in the direction $\boldsymbol{e}_1$.

Recall also that $\boldsymbol{e}_1$ is the direction of the greatest variation in the $\boldsymbol{u}_i$ and $\boldsymbol{e}_3$ the direction of the least variation.

For the hands data of Table 31.1, $\boldsymbol{e}_1$ points in the direction of the length of the left hand and $\boldsymbol{e}_3$ in the normal direction to the palm.

Thus the angle θ of the small rotation $\Phi_3(\theta\xi)$ is the most constrained when its axis ξ points in the direction of the least variation in the $\boldsymbol{u}_i$. θ is least constrained when ξ points in the direction of the greatest variation of the $\boldsymbol{u}_i$.

For the hands data, if $\widehat{\boldsymbol{h}}$ is in the direction of $\boldsymbol{e}_1$, the length of the hand, it represents a small rotation at the elbow with the wrist held rigid. The variance of the deviation rotation $\widehat{\boldsymbol{h}}$ in the direction $\boldsymbol{e}_1$ is $(\lambda_2+\lambda_3)^{-1} = 0.298$. If $\widehat{\boldsymbol{h}}$ points in the direction of $\boldsymbol{e}_2$, the width of the hand, it represents a forwards and backwards rotation at the wrist; the variance of $\widehat{\boldsymbol{h}}$ in this direction is $(\lambda_2+\lambda_3)^{-1} = 0.196$. Finally if $\widehat{\boldsymbol{h}}$ points in the direction of $\boldsymbol{e}_3$, the normal vector to the hand, it represents a somewhat awkward sideways rotation at the wrist (this rotation is represented in Fig. 31.1b; the variance of $\widehat{\boldsymbol{h}}$ in this direction is $(\lambda_1+\lambda_2)^{-1} = 0.121$. If the variability of the component of $\widehat{\boldsymbol{h}}$ in the direction of a rotation at the elbow

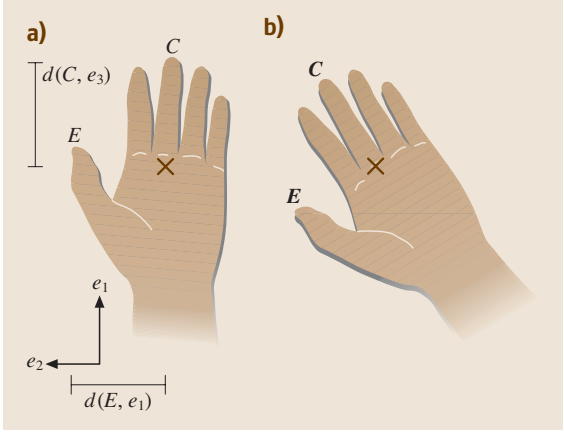


Fig. 31.1 **(a)**A hand with axes $\boldsymbol{e}_1, \boldsymbol{e}_2$; axis $\boldsymbol{e}_3$ points out of paper. X marks the center point $\overline{\boldsymbol{u}}$. The distances $d(C, \boldsymbol{e}_3)$ and $d(E, \boldsymbol{e}_1)$ are the lengths of the indicated line segments. **(b)**The effect of a rotation of angle θ around the axis $\boldsymbol{e}_3$. The point C moves a distance of approximately $d(C, \boldsymbol{e}_3)\theta$. Under a rotation of θ around $\boldsymbol{e}_1$ (not shown), the point E moves a distance of approximately $d(E, \boldsymbol{e}_1)\theta$. Notice that $d(E, \boldsymbol{e}_1) < d(C, \boldsymbol{e}_3)$, and, indeed, the landmarks $\boldsymbol{u}_i$ tend to be closer to $\boldsymbol{e}_1$ than to $\boldsymbol{e}_3$. It follows that a rotation of θ around $\boldsymbol{e}_3$ will move the figure more than a rotation of θ around $\boldsymbol{e}_1$

is unacceptably large, we need to increase λ_3; in effect to create, if possible, landmarks which effectively thicken the palm.

A heuristic derivation of this result is due to *Stock* and *Molnar* [31.8, 9]. It appeared in the geophysical literature and is considered a major development in our understanding of the uncertainties in tectonic plate reconstructions. We will present their argument below, suitably modified for the image registration context.

It is convenient to rewrite the model, as in Theorem 31.4.4, in the form (31.29). If we substitute $\mathbf{A} = \widehat{\mathbf{A}}\Phi_3(\theta\xi)$, we see that $\mathbf{A}$ first perturbs the $\boldsymbol{u}_i - \overline{\boldsymbol{u}}$ by the small rotation $\Phi_3(\theta\xi)$ and then applies the best fitting orthogonal matrix $\widehat{\mathbf{A}}$.

Let $d(\boldsymbol{u}_i, \xi)$ be the distance of the landmark $\boldsymbol{u}_i$ to the line through the center point $\overline{\boldsymbol{u}}$ and in the direction of the axis ξ. Refer to Fig. 31.1. Since the landmarks vary most in the direction $\boldsymbol{e}_1$ and least in the direction $\boldsymbol{e}_3$, the distances $d(\boldsymbol{u}_i, \boldsymbol{e}_3)$ will tend to be biggest and the distances $d(\boldsymbol{u}_i, \boldsymbol{e}_1)$ smallest.

A point $\boldsymbol{x}$ will move a distance of approximately $d(\boldsymbol{x}, \xi)\theta$ under a rotation of angle θ around the axis ξ. It follows that a rotation of angle θ will most move the landmarks $\boldsymbol{u}_i$ if the axis is $\boldsymbol{e}_3$. It will move the landmarks $\boldsymbol{u}_i$ least if the axis is $\boldsymbol{e}_1$. In other words, for a fixed θ, the small rotation $\Phi_3(\theta\xi)$ will most degrade the best fit, provided by $\widehat{\mathbf{A}}$, if $\xi = \boldsymbol{e}_3$; it will least degrade the best fit if $\xi = \boldsymbol{e}_1$.

An orthogonal transformation $\mathbf{A} = \widehat{\mathbf{A}}\Phi_3(\theta\xi)$ is considered a possible transformation if it does not degrade the best fit by too much. It follows that θ is most constrained if $\xi = \boldsymbol{e}_3$, the direction of the least variation in the landmarks $\boldsymbol{u}_i$, and is least constrained if $\xi = \boldsymbol{e}_1$, the direction of greatest variation in the landmarks $\boldsymbol{u}_i$.

Suppose instead we were to write the estimate $\widehat{\mathbf{A}}$ in the form

$$\widehat{\mathbf{A}} = \Phi_3(\widehat{\boldsymbol{h}}_v)\mathbf{A}\,, \qquad (31.46)$$
$$\mathbf{A} = \Phi_3(-\widehat{\boldsymbol{h}}_v)\widehat{\mathbf{A}}\,.$$

Then (31.43) is replaced by

$$\begin{aligned}\mathrm{Cov}(\widehat{\boldsymbol{h}}_v) = \frac{k}{n}\Big[&(\lambda_2+\lambda_3)^{-1}(\mathbf{A}\boldsymbol{e}_1)(\mathbf{A}\boldsymbol{e}_1)^{\mathrm{T}}\\ &+(\lambda_3+\lambda_1)^{-1}(\mathbf{A}\boldsymbol{e}_2)(\mathbf{A}\boldsymbol{e}_2)^{\mathrm{T}}\\ &+(\lambda_1+\lambda_2)^{-1}(\mathbf{A}\boldsymbol{e}_3)(\mathbf{A}\boldsymbol{e}_3)^{\mathrm{T}}\Big]\,.\end{aligned}$$

The same reasoning then expresses the errors of $\widehat{\boldsymbol{h}}_v$, and hence of $\widehat{\mathbf{A}}$, in terms of the geometry of the landmarks $\boldsymbol{v}_i$. In other words, for the hands data, using the definition (31.46) expresses the errors of $\widehat{\mathbf{A}}$ in terms of the orientation of the right hand.

31.4.9 Statistical Properties of *M*-Estimates for Spherical Regressions

The statistical assumptions of the spherical regression model (31.1) are:

- $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n \in \Omega_p$ are fixed (non-random) vectors.
- $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \Omega_p$ are independent random vectors.
- The distribution of $\boldsymbol{v}_i$ is of the form $f_0(t_i)$ where $t_i = \boldsymbol{v}_i^{\mathrm{T}}\mathbf{A}\boldsymbol{u}_i$. Here $\mathbf{A} \in \mathcal{SO}(p)$ or $\mathcal{O}(p)$ is unknown.

A commonly used distribution for spherical data $\boldsymbol{x} \in \Omega_p$ is the distribution whose density (with respect to surface measure, or uniform measure, on Ω_p) is

$$f(\boldsymbol{x};\theta) = c(\kappa)\exp(\kappa\boldsymbol{x}^{\mathrm{T}}\theta)\,. \qquad (31.47)$$

This distribution has two parameters: a positive real constant κ which is commonly called the *concentration parameter* and $\theta \in \Omega_p$. It is easily seen that $f(\boldsymbol{x})$ is maximized over $\boldsymbol{x} \in \Omega_p$ at θ and hence θ is usually refered to as the *modal vector*; $c(\kappa)$ is a normalizing constant.

If $\kappa = 0$, (31.47) is a uniform density on Ω_p. On the other hand as $\kappa \to \infty$, the density (31.47) approaches that of a multivariate normal distribution in $p-1$ dimensions with a covariance matrix of $\kappa^{-1}\mathbf{I}_{p-1}$. Thus intuitively we can think of κ as σ^{-2}, that is think of κ as the inverse variance. As $\kappa \to \infty$, (31.47) approaches a singular multivariate normal distribution supported on the $(p-1)$-dimensional subspace $\theta^{\perp} \subset \mathbb{R}^p$. As a singular multivariate normal distribution in $\mathbb{R}^p$ its covariance matrix is $\kappa^{-1}(\mathbf{I}_p - \theta\theta^{\mathrm{T}})$.

For the circle Ω_1, (31.47) is due to von Mises. For general Ω_p, it is due (independently) to Fisher and to Langevin. More properties of the Fisher–von Mises–Langevin distribution can be found in *Watson* [31.10] or in *Fisher* et al. [31.11].

The distribution of an M-estimator $\widehat{\mathbf{A}}$ which minimizes an objective function of the form (31.14) is similar to the distribution given in Theorem 31.4.4:

- If $p = 2$, write $\widehat{\mathbf{A}} = \mathbf{A}\Psi_2(\widehat{h})$, for $\widehat{h} \in \mathbb{R}^1$. Then $\widehat{h}$ is normally distributed with mean 0 and variance $\frac{k}{n}$.
- If $p = 3$, write $\widehat{\mathbf{A}} = \mathbf{A}\Psi_3(\widehat{\boldsymbol{h}})$, for $\widehat{\boldsymbol{h}} \in \mathbb{R}^3$. Let $\boldsymbol{\Sigma} = \lambda_1\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} + \lambda_2\boldsymbol{e}_2\boldsymbol{e}_2^{\mathrm{T}} + \lambda_3\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}}$ be the spectral decomposition of $\boldsymbol{\Sigma}$. Then $\widehat{\boldsymbol{h}}$ is distributed trivariate normal with mean $\mathbf{0}$ and covariance matrix

$$\begin{aligned}\frac{k}{n}\Big[&(\lambda_2+\lambda_3)^{-1}\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} + (\lambda_3+\lambda_1)^{-1}\boldsymbol{e}_2\boldsymbol{e}_2^{\mathrm{T}}\\ &+(\lambda_1+\lambda_2)^{-1}\boldsymbol{e}_3\boldsymbol{e}_3^{\mathrm{T}}\Big]\,.\end{aligned}$$

- For general p, write $\widehat{\mathbf{A}} = \mathbf{A}\exp(\widehat{\mathbf{H}})$, where $\widehat{\mathbf{H}}$ is $p \times p$ skew-symmetric. Then $\widehat{\mathbf{H}}$ has a multivariate normal density proportional to $\exp\left[-\frac{n}{2k}\mathrm{Tr}(\widehat{\mathbf{H}}^{\mathrm{T}}\boldsymbol{\Sigma}\widehat{\mathbf{H}})\right]$.

Let $\psi(t) = -\rho_0'(t)$. (The sign of ψ has been chosen to make $\psi(t)$ nonnegative, since ρ_0 is a decreasing function of t.) The constant k and its sample estimate $\widehat{k}$ are given by

$$k = \frac{(p-1)E[\psi(t)^2(1-t^2)]}{E^2[(p-1)\psi(t)t - \psi'(t)(1-t^2)]}\,,$$

$$\widehat{k} = \frac{n(p-1)\sum_i \psi(t_i)^2(1-t_i^2)}{\left\{\sum_i[(p-1)\psi(t_i)t_i - \psi'(t_i)(1-t_i^2)]\right\}^2}\,. \quad (31.48)$$

For the spherical case, the matrix $\boldsymbol{\Sigma} = \sum_i \boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}$. Its dominant eigenvector $\boldsymbol{e}_1$ points in the direction of the center of the $\boldsymbol{u}_i$. The $\boldsymbol{e}_2$ is the vector perpendicular to $\boldsymbol{e}_1$ so that the two-dimensional plane spanned by $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ (and the origin) best fits the $\boldsymbol{u}_i$. This continues until $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{p-1}$ is the $(p-1)$-dimensional hyperplane, among the collection of all $(p-1)$-dimensional hyperplanes that bests fits the data. This latter hyperplane is, of course, the hyperplane perpendicular to $\boldsymbol{e}_p$. Except for this slight reinterpretation of the geometric meaning of the $\boldsymbol{e}_i$, our previous comments about the relationship of the uncertainties in $\widehat{\boldsymbol{h}}$ to the geometry of the $\boldsymbol{u}$-points, as summarized by the eigen-decomposition of $\boldsymbol{\Sigma}$, remain valid. Indeed the original Stock and Molnar insights about the uncertainties of tectonic plate reconstructions were actually in the spherical data context.

Thus, as before, the uncertainties in $\widehat{\mathbf{A}}$ are determined up to the constant k by the geometry of the $\boldsymbol{u}$-points. Only the constant k depends upon the underlying data distribution f_0 or upon the objective function ρ. We can define the asymptotic relative efficiency as in Sect. 31.4.7 without change. Its interpretation (31.39) also remains valid.

Equation (31.48) implies that we can, as before, define the asymptotic efficiency of the L_1 estimator relative to the least squares estimator, at the density f_0, as $\mathrm{ARE}(L_1, L_2; f_0) = k(f_0, L_2)/k(f_0, L_1)$. The interpretation (31.39) remains valid. The constants $k(f_0, L_2)$ and $k(f_0, L_1)$ come from (31.48) using the underlying density f_0 under consideration and $\rho_0(t) = 2 - 2t$ [refer to (31.9)], $\psi(t) = 2$, for the least squares case, or $\rho_0(t) = \arccos(t)$, $\psi(t) = (1-t^2)^{\frac{1}{2}}$, for the L_1 case. If f_0 is the Fisher–von Mises–Langevin density (31.47) on Ω_p (which we will denote by $F_{\kappa,p}$ in the following)

$$\mathrm{ARE}(L_1, L_2; F_{\kappa,p}) = \frac{\left[\int_{-1}^{1} \mathrm{e}^{\kappa t}(1-t^2)^{(p-2)/2}\,\mathrm{d}t\right]^2}{\left[\int_{-1}^{1} \mathrm{e}^{\kappa t}(1-t^2)^{(p-1)/2}\,\mathrm{d}t\right]} \times \frac{1}{\left[\int_{-1}^{1} \mathrm{e}^{\kappa t}(1-t^2)^{(p-3)/2}\,\mathrm{d}t\right]}\,. \quad (31.49)$$

As $\kappa \to \infty$, the limit of (31.49) is

$$\lim_{\kappa\to\infty}\mathrm{ARE}(L_1, L_2; F_{\kappa,p}) = \frac{2\Gamma^2(p/2)}{(p-1)\Gamma^2[(p-1)/2]}\,. \quad (31.50)$$

Comparing (31.40) with (31.50), we see that (31.50) is the same as (31.40) with p replaced by $p-1$. This is as expected because, as noted above, for large κ the Fisher–von Mises–Langevin distribution approaches a $(p-1)$-dimensional multivariate normal distribution. Figure 31.2 gives a graph of $\mathrm{ARE}(L_1, L_2; F_{\kappa,p})$ for $p = 2, 3$.

In particular for $p = 3$, $\mathrm{ARE}(L_1, L_2; F_\kappa, 3) \to \pi/4$. For the Fisher–von-Mises–Langevin distribution, least squares methods are optimal. Nevertheless, in standard error terms, the penalty for using L_1 methods is at most 13%.

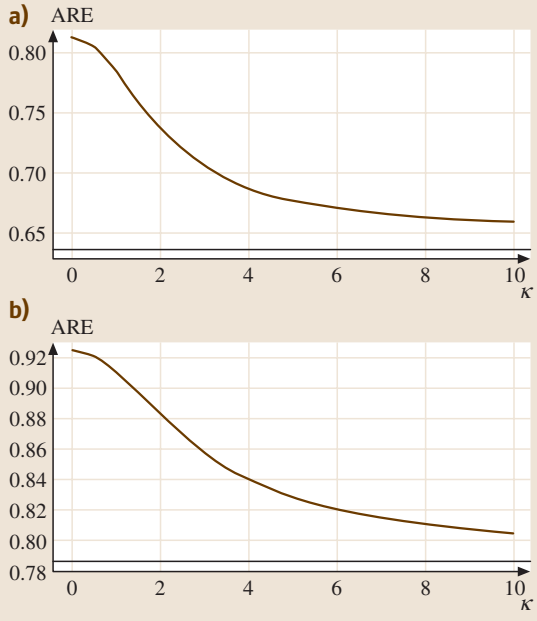


Fig. 31.2a,b Asymptotic efficiency of L_1 estimators relative to least squares estimators for Fisher–von Mises–Langevin distributions on Ω_p as a function of κ for **(a)** $p = 2$ and **(b)** $p = 3$. Horizontal lines are asymptotic limits as $\kappa \to \infty$.

31.5 Diagnostics

We discuss in this section influence function diagnostics for the Procrustes model. Suppose the registration provided by the estimates $(\widehat{\mathbf{A}}, \widehat{\gamma}, \widehat{\boldsymbol{b}})$ is unsatisfactory. These diagnostics will determine which points are influential for the estimated orthogonal matrix $\widehat{\mathbf{A}}$, which points are influential for the estimated scale change $\widehat{\gamma}$, and which are influential for the estimated translation $\widehat{\boldsymbol{b}}$.

31.5.1 Influence Diagnostics in Simple Linear Regression

As background discussion, we consider first the simple linear regression model

$$y_i = \alpha + \beta x_i + \text{error}\,, \tag{31.51}$$

where $x_i, y_i \in \mathbb{R}^1$. For simplicity, we will assume $\sum_i x_i = 0$. This can be accomplished by a centering transformation similar to that used in (31.29).

For the model (31.51), the least squares estimates are

$$\widehat{\alpha} = \overline{y}\,,$$
$$\widehat{\beta} = \left(\sum_i x_i^2\right)^{-1} \left(\sum_i x_i y_i\right)\,. \tag{31.52}$$

Suppose we delete the i-th observation (x_i, y_i) and recompute the estimates (31.52). The resulting estimates would be [see *Cook* and *Weisberg* [31.12], (3.4.6)]

$$\widehat{\alpha}_{(i)} = \widehat{\alpha} - (1 - v_{ii})^{-1} \frac{e_i}{n}\,,$$
$$\widehat{\beta}_{(i)} = \widehat{\beta} - (1 - v_{ii})^{-1} \frac{x_i e_i}{\sum_k x_k^2}\,, \tag{31.53}$$

where

$$e_i = y_i - \widehat{\alpha} - \widehat{\beta} x_i$$

is the residual, and

$$v_{ii} = \frac{1}{n} + \frac{x_i^2}{\sum_k x_k^2}$$

is the i-th diagonal entry of the so-called *hat* matrix. It can be shown that

$$0 \le v_{ii} \le 1\,,$$
$$\sum_i v_{ii} = 2\,. \tag{31.54}$$

If $|x_i|$ is big, $1 - v_{ii}$ can be close to zero, although because of (31.54), if n is large, this will usually not be the case. Ignoring the factor of $(1 - v_{ii})^{-1}$, it follows from (31.53) that deletion of (x_i, y_i) will be influential for $\widehat{\alpha}$ when the magnitude of the residual $|e_i|$ is big. Deletion of (x_i, y_i) will be influential for $\widehat{\beta}$ when both $|x_i|$ and $|e_i|$ are big. Points with large values of $|x_i|$ [typically, due to (31.54), $|x_i| > \frac{4}{n}$] are called *high-leverage* points, whereas points with large values of $|e_i|$ are called *outliers*. (Recall we have centered the data so that $\overline{x} = 0$.)

Thus influence on $\widehat{\alpha}$ and on $\widehat{\beta}$ are different. Outliers are influential for $\widehat{\alpha}$, whereas influence for $\widehat{\beta}$ is a combination of being an outlier and having high leverage. For the model (31.51) with the least squares estimators, the *influence function* works out to be

$$\mathrm{IF}\left[\widehat{\alpha}; (x_i, y_i)\right] = \frac{y_i - \alpha - \beta x_i}{n}\,, \tag{31.55}$$

$$\mathrm{IF}\left[\widehat{\beta}; (x_i, y_i)\right] = \frac{x_i(y_i - \alpha - \beta x_i)}{\sum_k x_k^2}\,, \tag{31.56}$$

where α and β are the 'true' population values in the model (31.51). We will not give a formal definition of the influence function here, but refer the reader to *Cook* and *Weisberg* [31.12] for a more comprehensive discussion of the influence function in the regression model.

It should be noted tthat to actually calculate (31.55) and (31.56) from a sample, it is necessary to estimate α and β. Thus, even though in the left-hand sides of (31.55) and (31.56), $\widehat{\alpha}$ and $\widehat{\beta}$ are least squares estimates, we should substitute in the right-hand sides of (31.55) and (31.56) better estimates, if available, of α and β. There is no contradiction in using L_1 estimates to estimate the influence function of the least squares estimators.

31.5.2 Influence Diagnostics for the Procrustes Model

Chang and *Ko* [31.3] calculated the *standardized influence functions* (SIF) for M-estimates (31.13) in the Procrustes model (31.29). (The influence functions of the estimates $\widehat{\mathbf{A}}$ and $\widehat{\beta}$ are vectors; the standardization calculates their square length in some metric.) Using their notation

$$||\mathrm{SIF}\left[\widehat{\beta}; (\boldsymbol{u}_i, \boldsymbol{v}_i)\right]||^2 = k_I \psi(s_i)^2\,, \tag{31.57}$$

where $s_i = ||\boldsymbol{v}_i - \gamma \mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \beta||$ and $\psi(s) = \rho_0'(s)$. Therefore

- the influence of $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ on the estimate $\widehat{\beta}$ of the translation parameter depends only upon the length s_i of the residual.

This behavior is similar to that of simple linear regression (31.55). The constant k_I is given by

$$k_I = \frac{pE[g'(s)^2]}{E^2[\psi'(s)+(p-1)\psi(s)s^{-1}]} ,$$

where $g(s) = \log f_0(s)$ and $f_0(s)$ is defined in Sect. 31.4.3.

For the scale parameter γ, let $\mathbf{\Sigma} = n^{-1}\sum_i (\boldsymbol{u}_i - \overline{\boldsymbol{u}})(\boldsymbol{u}_i - \overline{\boldsymbol{u}})^{\mathrm{T}}$. Then

$$||\mathrm{SIF}[\widehat{\gamma}; (\boldsymbol{u}_i, \boldsymbol{v}_i)]||^2 = \frac{k_I \psi(s_i)^2}{\mathrm{Tr}(\mathbf{\Sigma})} \left[\boldsymbol{w}_i^{\mathrm{T}} \mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}})\right]^2 . \quad (31.58)$$

Here

$$\boldsymbol{w}_i = [\boldsymbol{v}_i - \gamma \mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \beta]/s_i .$$

Notice that $\boldsymbol{v}_i - \gamma\mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \beta$ is the residual of the i-th data point and s_i is its length. Thus $\boldsymbol{w}_i$ is a unit-length vector in the direction of the i-th data point. We conclude

- For a given length s_i of residual, a point $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ will be influential for the estimate $\widehat{\gamma}$ of the scale parameter if $\boldsymbol{u}_i$ is far from the center $\overline{\boldsymbol{u}}$ of the data and if its residual is parallel to $\mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}})$.

For simplicity, we restrict the formulas of influence on the estimate of the orthogonal matrix $\mathbf{A}$ to the cases $p = 2, 3$. For $p = 2$,

$$||\mathrm{SIF}\left[\widehat{\mathbf{A}}; (\boldsymbol{u}_i, \boldsymbol{v}_i)\right]||^2 = \frac{k_I \psi(s_i)^2}{\mathrm{Tr}(\mathbf{\Sigma})} ||\boldsymbol{w}_i \times [\mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}})]||^2 . \quad (31.59)$$

The product on the right-hand side of (31.59) is the vector 'cross' product. Therefore

- For $p = 2$, for a given length s_i of residual, a point $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ will be influential for the estimate $\widehat{\mathbf{A}}$ of the orthogonal matrix if $\boldsymbol{u}_i$ is far from the center $\overline{\boldsymbol{u}}$ of the data and if its residual is perpendicular to $\mathbf{A}(\boldsymbol{u}_i - \overline{\boldsymbol{u}})$. Thus points which are influential for $\widehat{\mathbf{A}}$ will not be influential for $\widehat{\gamma}$, and vice versa. Indeed

$$||\mathrm{SIF}[\widehat{\gamma}; (\boldsymbol{u}_i, \boldsymbol{v}_i)]||^2 + ||\mathrm{SIF}\left[\widehat{\mathbf{A}}; (\boldsymbol{u}_i, \boldsymbol{v}_i)\right]||^2 = \frac{k_I \psi(s_i)^2}{\mathrm{Tr}(\mathbf{\Sigma})} ||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||^2 .$$

For $p = 3$, let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the eigenvalues of Σ and let $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ be the corresponding eigenvectors. Write

$$\boldsymbol{w}_i \times \left(\mathbf{A}\frac{\boldsymbol{u}_i - \overline{\boldsymbol{u}}}{||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||}\right) = x_1 \mathbf{A}\boldsymbol{e}_1 + x_2 \mathbf{A}\boldsymbol{e}_2 + x_3 \mathbf{A}\boldsymbol{e}_3 .$$

Then

$$\mathrm{SIF}\left[\widehat{\mathbf{A}}; (\boldsymbol{u}_i, \boldsymbol{v}_i)\right] = k_I \psi(s_i)^2 ||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||^2 \times \left(\frac{x_1^2}{\lambda_2 + \lambda_3} + \frac{x_2^2}{\lambda_3 + \lambda_1} + \frac{x_3^2}{\lambda_1 + \lambda_2}\right) . \quad (31.60)$$

It follows

- For $p = 3$, for a given length s_i of residual and distance $||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||$ of $\boldsymbol{u}_i$ from the center of the data, a point $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ will be maximally influential for the estimate $\widehat{\mathbf{A}}$ of the orthogonal matrix if both $\boldsymbol{u}_i - \overline{\boldsymbol{u}}$ is perpendicular to the dominant eigenvector $\boldsymbol{e}_1$ of $\mathbf{\Sigma}$ and the residual

$$\boldsymbol{w}_i = \pm\mathbf{A}\left(\frac{\boldsymbol{u}_i - \overline{\boldsymbol{u}}}{||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||} \times \boldsymbol{e}_1\right) .$$

- The influence of $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ on $\widehat{\mathbf{A}}$ will be zero if

$$\boldsymbol{w}_i = \pm\mathbf{A}\left(\frac{\boldsymbol{u}_i - \overline{\boldsymbol{u}}}{||\boldsymbol{u}_i - \overline{\boldsymbol{u}}||}\right) .$$

- The maximum influence of the data on the estimate $\widehat{\mathbf{A}}$ of the orthogonal matrix can be minimized for fixed $\mathrm{Tr}(\mathbf{\Sigma})$ by making $\lambda_1 = \lambda_2 = \lambda_3$. Thus the optimal choice of landmarks would make the landmarks spherically symmetric around the center point $\overline{\boldsymbol{u}}$.

31.5.3 Example: Influence for the Hands Data

For the Procrustes model (31.29) and the hands data, we compare here the influence statistics for the least squares estimates $(\widehat{\mathbf{A}}_2, \widehat{\gamma}_2, \widehat{\boldsymbol{\beta}}_2)$ [given in (31.20) and (31.33)] to those for the the L_1 estimates $(\widehat{\mathbf{A}}_1, \widehat{\gamma}_1, \widehat{\boldsymbol{\beta}}_1)$ [in (31.34)]. These estimates correspond to $\psi_2(s) = s$ and $\psi_1(s) = 1$ respectively. In the right-hand sides of (31.57), (31.58), and (31.60), we substituted $s_i = ||\boldsymbol{v}_i - \widehat{\gamma}_1 \widehat{\mathbf{A}}_1(\boldsymbol{u}_i - \overline{\boldsymbol{u}}) - \widehat{\boldsymbol{\beta}}_1||$ to calculate the influence functions for both the L_1 and least squares estimates. Similarly the $\boldsymbol{w}_i$ were calculated using the L_1 estimates. Furthermore when (31.57), (31.58), and (31.60) were calculated for the i-th observation $(\boldsymbol{u}_i, \boldsymbol{v}_i)$, $\boldsymbol{u}_i$ was not used to calculate $\mathbf{\Sigma}$.

Using (31.57),

$$||\mathrm{SIF}[\widehat{\boldsymbol{\beta}}_2; (\boldsymbol{u}_i, \boldsymbol{v}_i)]||^2 \propto s_i^2 ,$$
$$||\mathrm{SIF}[\widehat{\boldsymbol{\beta}}_1; (\boldsymbol{u}_i, \boldsymbol{v}_i)]||^2 \propto 1 ,$$

so that E (top of thumb), followed by H (base of palm), are the most influential for $\widehat{\beta}_2$. All points are equally influential for $\widehat{\beta}_1$.

In what follows we will be interested in determining which data points are most influential for which estimates. In other words we will be interested in the relative values of $||\mathrm{SIF}||^2$. Thus, for each estimator, we renormalized the values of $||\mathrm{SIF}||^2$ so that their sum (over the 12 data points) equals 1. The results, together with the values of s_i, are shown in Fig. 31.3.

We have from (31.58) and (31.60)

$$||\mathrm{SIF}\,[\widehat{\gamma}_2;(\boldsymbol{u}_i,\boldsymbol{v}_i)]\,||^2 \propto s_i^2\left[\boldsymbol{w}_i^{\mathrm{T}}\mathbf{A}(\boldsymbol{u}_i-\overline{\boldsymbol{u}})\right]^2 ,$$

$$||\mathrm{SIF}\,[\widehat{\gamma}_1;(\boldsymbol{u}_i,\boldsymbol{v}_i)]\,||^2 \propto \left[\boldsymbol{w}_i^{\mathrm{T}}\mathbf{A}(\boldsymbol{u}_i-\overline{\boldsymbol{u}})\right]^2 ,$$

$$\left|\left|\mathrm{SIF}\left[\widehat{\mathbf{A}}_2;(\boldsymbol{u}_i,\boldsymbol{v}_i)\right]\right|\right|^2 \propto s_i^2\left|\left|\boldsymbol{u}_i-\overline{\boldsymbol{u}}\right|\right|^2 \left(\frac{x_1^2}{\lambda_2+\lambda_3}+\frac{x_2^2}{\lambda_3+\lambda_1}+\frac{x_3^2}{\lambda_1+\lambda_2}\right) ,$$

$$\left|\left|\mathrm{SIF}\left[\widehat{\mathbf{A}}_1;(\boldsymbol{u}_i,\boldsymbol{v}_i)\right]\right|\right|^2 \propto \left|\left|\boldsymbol{u}_i-\overline{\boldsymbol{u}}\right|\right|^2 \left(\frac{x_1^2}{\lambda_2+\lambda_3}+\frac{x_2^2}{\lambda_3+\lambda_1}+\frac{x_3^2}{\lambda_1+\lambda_2}\right) ,$$

$$\boldsymbol{w}_i\times\left[\widehat{\mathbf{A}}_1\frac{\boldsymbol{u}_i-\overline{\boldsymbol{u}}}{||\boldsymbol{u}_i-\overline{\boldsymbol{u}}||}\right] = x_1\widehat{\mathbf{A}}_1\boldsymbol{e}_1+x_2\widehat{\mathbf{A}}_1\boldsymbol{e}_2+x_3\widehat{\mathbf{A}}_1\boldsymbol{e}_3 .$$

Examining Fig. 31.3, we see that point E is by far the most influential point for $\widehat{\mathbf{A}}$. Its relative influence however can be somewhat diminished by using L_1 estimates. The value of $||\boldsymbol{u}_{\mathrm{E}}-\overline{\boldsymbol{u}}||$ is also the largest of the $||\boldsymbol{u}_i-\overline{\boldsymbol{u}}||$. It turns out that $\boldsymbol{u}_{\mathrm{E}}-\overline{\boldsymbol{u}}$ makes an angle of 13° with $\boldsymbol{e}_2$ and that the unit length $\boldsymbol{w}_{\mathrm{E}}$ makes an angle of 12° with $\widehat{\mathbf{A}}_1\boldsymbol{e}_3$. Thus x_1 will be relatively big and x_2, x_3 relatively small. This accounts for the strong influence of point E on both estimates of $\mathbf{A}$. Notice that s_{E} and $||\boldsymbol{u}_{\mathrm{E}}-\overline{\boldsymbol{u}}||$ are sufficiently big that, despite the directions of $\boldsymbol{w}_{\mathrm{E}}$ and $\widehat{A}_1(\boldsymbol{u}_{\mathrm{E}}-\overline{\boldsymbol{u}})$, E is still fairly influential for $\widehat{\gamma}_2$. However, its influence on $\widehat{\gamma}_1$, which does not depend upon s_{E}, is quite small.

The point H (base of the palm) is the most influential point for $\widehat{\gamma}$. H is perhaps the least well-defined point so that it is not surprising that its residual length s_{H} is relatively big. It also defines the length of the hand, so that its influence on $\widehat{\gamma}$ is not surprising. Indeed if H were completely deleted, $\widehat{\gamma}_2$ would change from 0.9925 to 1.0110 and $\widehat{\gamma}_1$ changes from 1.0086 to 1.0262.

One might think that C (top of the middle finger) would also be influential for $\widehat{\gamma}$. In a coordinate system of the eigenvectors of $\boldsymbol{\Sigma}$,

$$\boldsymbol{u}_{\mathrm{H}}-\overline{\boldsymbol{u}} = \begin{bmatrix} -3.98 & 1.15 & 0.33 \end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{u}_{\mathrm{C}}-\overline{\boldsymbol{u}} = \begin{bmatrix} 3.55 & -0.77 & -0.03 \end{bmatrix}^{\mathrm{T}}$$

so that $\boldsymbol{u}_{\mathrm{H}}-\overline{\boldsymbol{u}} \approx -(\boldsymbol{u}_{\mathrm{C}}-\overline{\boldsymbol{u}})$. It is useful here to remember that $\boldsymbol{e}_1$ is approximately in the direction of the length of the left hand. Furthermore s_{C} and s_{H} are reasonably close.

However Fig. 31.3 indicates that C has negligible influence on both estimates of γ. Indeed if C were completely deleted, $\widehat{\gamma}_2$ would only change from 0.9925 to 0.9895 and $\widehat{\gamma}_1$ change from 1.0086 to 1.0047. These changes are much smaller than those caused by the deletion of H.

The difference is that $\widehat{\mathbf{A}}_1(\boldsymbol{u}_{\mathrm{C}}-\overline{\boldsymbol{u}})$ makes an angle of 88° with $\boldsymbol{w}_{\mathrm{C}}$. In other words $\widehat{\mathbf{A}}_1(\boldsymbol{u}_{\mathrm{C}}-\overline{\boldsymbol{u}})$ and $\boldsymbol{w}_{\mathrm{C}}$ are very close to perpendicular (Perhaps the close to perpendicularity of the residual at C to $\widehat{\mathbf{A}}_1(\boldsymbol{u}_{\mathrm{C}}-\overline{\boldsymbol{u}})$ is to be expected. The uncertainty in locating C is roughly tangential to the middle finger tip.) Hence the influence of C on $\widehat{\gamma}$ is negligible.

On the other hand, $\widehat{\mathbf{A}}_1(\boldsymbol{u}_{\mathrm{H}}-\overline{\boldsymbol{u}})$ makes an angle of 124° with $\boldsymbol{w}_{\mathrm{H}}$. This accounts for the greater influence of H.

Thus if the registration between the two hands is unsatisfactory in either the translation or rotation parameters, point E should be inspected. If it is unsatisfactory in the scale change, point H should be checked.

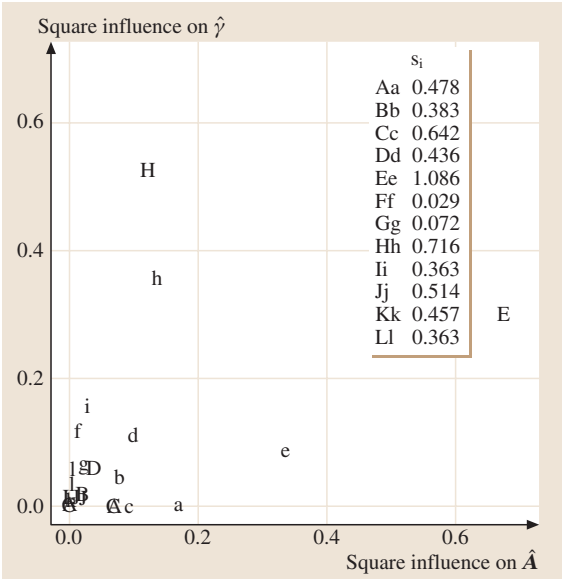


Fig. 31.3 Relative influence of hands data points on least squares (upper case) and L_1 estimates (lower case) of γ and $\mathbf{A}$

References

31.1 G. Wahba: Section on problems and solutions: A least squares estimate of satellite attitude, SIAM Rev. **8**, 384–385 (1966)

31.2 G.R. Chapman, G. Chen, P.T. Kim: Assessing geometric integrity through spherical regression techniques, Stat. Sin. **5**, 173–220 (1995)

31.3 T. Chang, D. Ko: *M*-estimates of rigid body motion on the sphere and in Euclidean space, Ann. Stat. **23**, 1823–1847 (1995)

31.4 Colin, Goodall: Procrustes methods in the statistical analysis of shape, J. R. Stat. Soc. B **53**, 285–339 (1991)

31.5 T. Chang: Spherical regression, Ann. Stat. **14**, 907–924 (1986)

31.6 P.J. Huber: *Robust Statistics* (Wiley, New York 1981)

31.7 H. Goldstein: *Classical Mechanics* (Addison–Wesley, Reading 1950)

31.8 J. Stock, P. Molnar: Some geometrical aspects of uncertainties in combined plate reconstructions, Geology **11**, 697–701 (1983)

31.9 P. Molnar, J. Stock: A method for bounding uncertainties in combined plate reconstructions, J. Geophys. Res. **90**, 12537–12544 (1985)

31.10 G.S. Watson: *Statistics on Spheres* (Wiley Interscience, New York 1983)

31.11 N.I. Fisher, T. Lewis, B.J.J. Embleton: *Statistical Analysis of Spherical Data* (Cambridge Univ. Press, Cambridge 1987)

31.12 R.D. Cook, S. Weisberg: *Residuals and Influence in Regression* (Chapman Hall, New York 1982)

32. Statistical Genetics for Genomic Data Analysis

In this chapter, we briefly summarize the emerging statistical concepts and approaches that have been recently developed and applied to the analysis of genomic data such as microarray gene expression data. In the first section we introduce the general background and critical issues in statistical sciences for genomic data analysis. The second section describes a novel concept of statistical significance, the so-called false discovery rate, the rate of false positives among all positive findings, which has been suggested to control the error rate of numerous false positives in large screening biological data analysis. In the next section we introduce two recent statistical testing methods: significance analysis of microarray (SAM) and local pooled error (LPE) tests. The latter in particular, which is significantly strengthened by pooling error information from adjacent genes at local intensity ranges, is useful to analyze microarray data with limited replication. The fourth section introduces analysis of variation (ANOVA) and heterogenous error modeling (HEM) approaches that have been suggested for analyzing microarray data obtained from multiple experimental and/or biological conditions. The last two sections describe data exploration and discovery tools largely termed *supervised learning* and *unsupervised learning*. The former approaches include several multivariate statistical methods for the investigation of coexpression patterns of multiple genes, and the latter approaches are used as classification methods to discover genetic markers for predicting important subclasses of human diseases. Most of the statistical software packages for the approaches introduced in this chapter are freely available at the open-source bioinformatics software web site (Bioconductor; http://www.bioconductor.org/).

Accelerated by the Human Genome Project, recent advances in high-throughput biotechnologies have dramatically changed the horizon of biological and biomedical sciences. Large screening expression profiling techniques such as DNA microarrays, mass spectrometry, and protein chips offer great promise for functional genomics and proteomics research, and have the potential to transform the diagnosis and treatment of human diseases [32.1]. In particular, DNA microarray and GeneChip^{TM}gene expression approaches are becoming increasingly important in current biomedical studies [32.2–4].

Analysis of such genome-wide data, however, has brought extreme challenges not only in the biological sciences but also in the statistical sciences. Fundamental difficulties exist in applying traditional statistical approaches to genome-wide expression data, namely the *multiple comparisons issue* and the *small n–large p problem* [32.5]. The former problem arises because classical statistical hypothesis testing, modeling, and inference strategies are designed for studying a small number of candidate targets at a time, whereas one often investigates tens of thousands of genes' differential expression in a single microarray study. For example,

when a two-sample t-test is applied for evaluating statistical significance of thousands of genes' differential expression patterns in a microarray study, the p-values obtained from this within-gene test must be adjusted to take into account the random chance of all the candidate genes in the array data.

The latter difficulty, the small n–large p problem, arises due to the fact that many biological and biomedical microarray studies are performed with a small number of replicated arrays, or without replication. Unlike DNA sequence information, gene expression data are context-dependent and offer different interpretations depending on (patient) sample condition, time point, and treatment for a single subject [32.6]. In addition to the high costs of microarray experiments, certain biological or human patient specimens are often limited, thereby necessitating that microarray studies be performed with limited replication. Consequently, one must perform statistical inference on a small number of observations (n) compared to a large number of potential predictor genes (p). The latter number of tens of thousands of genes is simply too large to be considered in standard statistical testing and modeling, whereas the sample size (or number of replicated arrays at each condition) of a microarray study is typically small, a few tens at most and often only one or two replicates. This presents great difficulty for the application of traditional statistical approaches, which generally require a reasonably large sample size for maximal performance. As microarray (and similar high-throughput) technology becomes an important tool in biological and biomedical investigation, the lack of appropriate statistical methods for large screening microarray data will undoubtedly become a great obstacle in the current biological sciences. In this chapter, we will briefly summarize the statistical approaches that have been applied to microarray gene expression data analysis by avoiding these pitfalls.

We also introduce several multivariate statistical methods that have been used to investigate the coregulation structures of multiple genes as *unsupervised learning* and to discover genetic markers for predicting important subclasses of human diseases and biological targets as *supervised learning*. In particular, clustering approaches have been widely applied to the analysis of gene expression microarray data. The method of visualizing gene expression data based on cluster order, so-called cluster-image map (CIM) analysis, is found to be very efficient in summarizing the thousands of gene expression values and aiding in the identification of some interesting patterns of gene expression [32.3,7,8]. Since a clustering algorithm provides an efficient dimension reduction for extremely high-dimensional data based on their association, it is much easier to simultaneously screen thousands of gene expression values and to identify interesting patterns on the image maps. The statistical software packages for most of these approaches are freely available at the open-source bioinformatics development web site (Bioconductor; http://www.bioconductor.org/).

We note that this kind of microarray data analysis is implemented based on certain standard preprocessing procedures. Suppose we have gene expression data with n genes and p arrays. A matrix of this gene expression data is defined by $Y_{n\times p}$ with n rows and p columns. The data are then typically log2-transformed to remedy the right-skewed distribution, to make error components additive, and to aplly other statistical procedures that are based on underlying Gaussian distributional assumptions. Each column of the matrix (or each array) is scaled or normalized to a common baseline by matching interquartile ranges or by nonparametric regression methods, e.g., lowess.

32.1 False Discovery Rate

In order to avoid a large number of false positive findings (or type I errors) in genomic data analysis, the classical family-wise error rate (FWER) has initially been used to control for the random chance of multiple candidates by evaluating the probability that at most one false positive is included at a cutoff level of a statistic [32.9]. However, FWER adjustment has been found to be very conservative in microarray studies, resulting in a high false-negative error rate [32.10]. To avoid pitfalls such as this, a novel statistical significance concept, the so-called *false discovery rate* (FDR) and its refinement, the *q-value*, have been suggested [32.11, 12] (QVALUE package at Bioconductor). FDR is defined as follows. Consider a family of m simultaneously tested null hypotheses of which m_0 are true. For each hypothesis H_i a test statistic is calculated along with the corresponding p-value, P_i. Let R denote the number of hypotheses rejected by a procedure, V the number of true null

hypotheses erroneously rejected, and S the number of false hypotheses rejected as summarized in Table 32.1. Now let Q denote V/R when $R > 0$ and 0 otherwise. Then the FDR is defined as the expectation of Q, i. e. FDR$= E(Q)$.

As shown in [32.11], the FDR of a multiple comparison procedure is always smaller than or equal to the FWER, where equality holds if all null hypotheses are true. Thus, control of the FDR implies control of the FWER only when all null hypotheses are true, but it generally controls such an error rate much less conservatively than FWER because there exist quite a few true positives in practical data analysis. In the context of gene expression analysis, this result means that, if FDR is controlled at some level q, then the probability of erroneously detecting any differentially expressed genes among all genes identified by a certain selection criterion is less than or equal to q. Intuitively, FDR controls the expected proportion of false positives among all candidate genes identified significantly by a testing criterion. Therefore, based on FDR, researchers can now assess their statistical confidence among the identified targets with a much smaller false-negative error rate. FDR evaluation has been rapidly adopted for microarray data analysis including the significance analysis of microarrays (SAM) and local pooled error (LPE) approaches [32.9, 10, 13].

Table 32.1 Outcomes when testing m hypotheses

Hypothesis	Accept	Reject	Total
Null true	U	V	m_0
Alternative true	T	S	m_1
Total	W	R	m

32.2 Statistical Tests for Genomic Data

Each gene's differential expression pattern in a microarray experiment is usually assessed by (typically pairwise) contrasts of mean expression values among experimental conditions. Such comparisons have been routinely assessed as fold changes whereby genes with greater than two or three fold changes are selected for further investigation. It has been frequently found that a gene showing a high fold-change between experimental conditions might also exhibit high variability and hence its differential expression may not be significant. Similarly, a modest change in gene expression may be significant if its differential expression pattern is highly reproducible. A number of authors have pointed out this fundamental flaw in the fold-change-based approach [32.14]. In order to assess differential expression in a way that controls both false positives and false negatives, a standard approach is emerging based on statistical significance and hypothesis testing, with careful attention paid to the reliability of variance estimates and multiple comparison issues.

The classical two-sample t-statistic has initially been used for testing each gene's differential expression; procedures such as the Westfall–Young step-down method have been suggested to control FWER [32.9]. These t-test procedures, however, rely on reasonable estimates of reproducibility or within-gene error to be constructed, requiring a large number of replicated arrays. When a small number of replicates are available per condition, e.g. duplicate or triplicate, the use of naive, within-gene estimates of variability does not provide a reliable hypothesis-testing framework. For example, a gene may have very similar differential expression values in duplicate experiments by chance alone. This can lead to inflated signal-to-noise ratios for genes with low but similar expression values. Furthermore, the comparison of means can be misled by outliers with dramatically smaller or bigger expression intensities than other replicates. As such, error estimates constructed solely within genes may result in underpowered tests for differential expression comparisons and also result in large numbers of false positives.

A number of approaches to improving estimates of variability and statistical tests of differential expression have thus recently emerged. Several variance function methods have been proposed. Reference [32.15] suggested a simple regression estimation of local variances; [32.16] used a smoothing-spline pooled-error method by regressing standard error estimates on the mean log intensities; and [32.17] estimates a two-parameter variance function of mean expression intensity. Reference [32.18] compared some of these variance-estimation methods. Recently, [32.19] suggested the use of data-adapted robust estimate of array error based on a smoothing spline and standardized local median absolute deviation (MAD). The variance function methods described above borrow strength across genes in order to improve reliability of variance estimates in differential expression tests. This is conceptually similar to the SAM method of [32.10] and the empirical Bayes methods of [32.20] and [32.21].

These methods also shrink the within-gene variance estimate towards an estimate including more genes, and construct signal-to-noise ratios using the shrunken variance in a similar fashion to the local-pooled-error (LPE) test described below.

The local-pooled-error (LPE) estimation strategy has also been introduced for within-gene expression error, whereby variance estimates for genes are formed by pooling variance estimates for genes with similar expression intensities from replicated arrays within experimental conditions [32.13]. The LPE approach leverages the observations that genes with similar expression intensity values often show similar array-experimental variability within experimental conditions; and that variance of individual gene expression measurements within experimental conditions typically decreases as a (nonlinear) function of intensity [32.5,22]. The LPE approach handles the situation where a gene with low expression may have very low variance by chance and the resulting signal-to-noise ratio is unrealistically large. The pooling of errors within local intensities shrinks such variances to the variance of genes with similar intensities. In this chapter, two recent statistical testing procedures – SAM and LPE – are described in more detail while many classical testing and p-value adjustment strategies can be found elsewhere [32.9].

32.2.1 Significance Analysis of Microarrays

The significance analysis of microarrays (SAM) approach is based on analysis of random fluctuations in the data [32.10] (SIGGENES package at Bioconductor). Based on the observation that the signal-to-noise ratio decreases with decreasing gene expression, as shown in [32.13], fluctuations are considered to be gene specific even for a given level of expression in [32.10]. To account for gene-specific fluctuations, a statistic is defined based on the ratio of change in gene expression to standard deviation in the data for that gene. The *relative difference* $d(i)$ in gene expression is:

$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}\,, \tag{32.1}$$

where $x_I(i)$ and $x_U(i)$ are defined as the average levels of expression for gene (i) in states I and U, respectively. The *gene-specific scatter* $s(i)$ is the standard deviation of repeated expression measurements:

$$s(i) = \sqrt{a\left\{\sum_m [x_m(i) - \bar{x}_I(i)]^2 + \sum_n [x_n(i) - \bar{x}_U(i)]^2\right\}}\,, \tag{32.2}$$

where $\sum_m$ and $\sum_n$ are summations of the expression measurements in states I and U, respectively, $a = (1/n_1 + 1/n_2)/(n_1 + n_2 - 2)$, and n_1 and n_2 are the numbers of measurements in states I and U. To compare values of $d(i)$ across all genes, the distribution of $d(i)$ is assumed to be independent of the level of gene expression. At low expression levels, variance in $d(i)$ can be high because of small values of $s(i)$. To ensure that the variance of $d(i)$ is independent of gene expression, a small positive constant s_0 is added to the denominator of (32.1). The coefficient of variation of $d(i)$ is computed as a function of $s(i)$ in moving windows across the data. The value for s_0 was chosen to minimize the coefficient of variation.

32.2.2 The Local-Pooled-Error Test

The local-pooled-error (LPE) method has been introduced specifically for analysis of small-sample microarray data, whereby error variance estimates for genes are formed by pooling variance estimates for genes with similar expression intensities from replicated arrays within experimental conditions [32.13] (LPE package at Bioconductor). The LPE approach leverages the observations that genes with similar expression intensity values often show similar array-experimental variability within experimental conditions; and that variance of individual gene-expression measurements within experimental conditions typically decreases as a (nonlinear) function of intensity, as shown in Fig. 32.1. This is due, in part, to common background noise at each spot of the microarray. At high levels of expression intensity, this background noise is dominated by the expression intensity, while at low levels the background noise is a larger component of the observed expression intensity. The LPE approach controls the situation where a gene with low expression may have very low variance by chance and the resulting signal-to-noise ratio is unrealistically large. The LPE method borrows strength across genes in order to improve accuracy of error variance estimation in microarray data. This is conceptually similar to the SAM method above and the empirical Bayes methods of [32.20], which shrink the within-gene variance estimate towards an estimate including more genes in a similar fashion to LPE.

To take into account heterogeneous error variability across different intensity ranges in microarray data, the LPE method can be applied as follows (refer to [32.13] for a more detailed technical description). For oligo array data, let x_{ijk} be the observed expression intensity at gene j for array k and sample i. For duplicate

arrays, $k = 1, 2$, plots of $A = \log 2(x_{ij_1} x_{ij_2})/2$ versus $M = \log 2(x_{ij_1}/x_{ij_2})$, $j = 1, \ldots, J$, can facilitate the investigation of between-duplicate variability in terms of overall intensity. The A versus M (or AM) plot provides a very raw look at the data and is useful in detecting outliers and patterns of intensity variation as a function of mean intensity [32.9]. At each of the local intensity regions of the AM plot under a particular biological condition, the unbiased estimate of the local variance is obtained. A cubic spline is then fit to these local variance estimates to obtain a smoothing variance function. The optimal choice of the effective degree of freedom df_λ of the fitted smoothing spline is obtained by minimizing the expected squared prediction error. This two-stage error estimation approach – estimation of the error of M within quantiles and then nonparametric smoothing on these estimates – is used because direct nonparametric estimation often leads to unrealistic (small or large) estimates of error when only a small numbers of observations are available at a fixed-width intensity range.

Based on the LPE estimation above, statistical significance of the LPE-based test is evaluated as follows. First, each gene's medians under the two compared conditions are calculated to avoid artifacts from outliers. The approximate normality of medians can be assumed with a small number of replicates based on the fact that the individual log-intensity values within a local intensity range follow a normal distribution [32.13]. The LPE statistic for the median (log-intensity) difference z is then calculated as:

$$z = (m_1 - m_2)/s_{\text{pooled}} , \qquad (32.3)$$

where m_1 and m_2 are the median intensities in two the compared array-experimental conditions X and Y, and s_{pooled} is the pooled standard error, $\left[s_x^2(m_1)/n_1 + s_y^2(m_2)/n_2\right]^{1/2}$ from the LPE-estimated baseline variances of s_x^2 and s_y^2. The LPE approach shows a significantly better performance than two-sample t-test, SAM, and Westfall–Young permutation tests, especially when the number of replicates is smaller

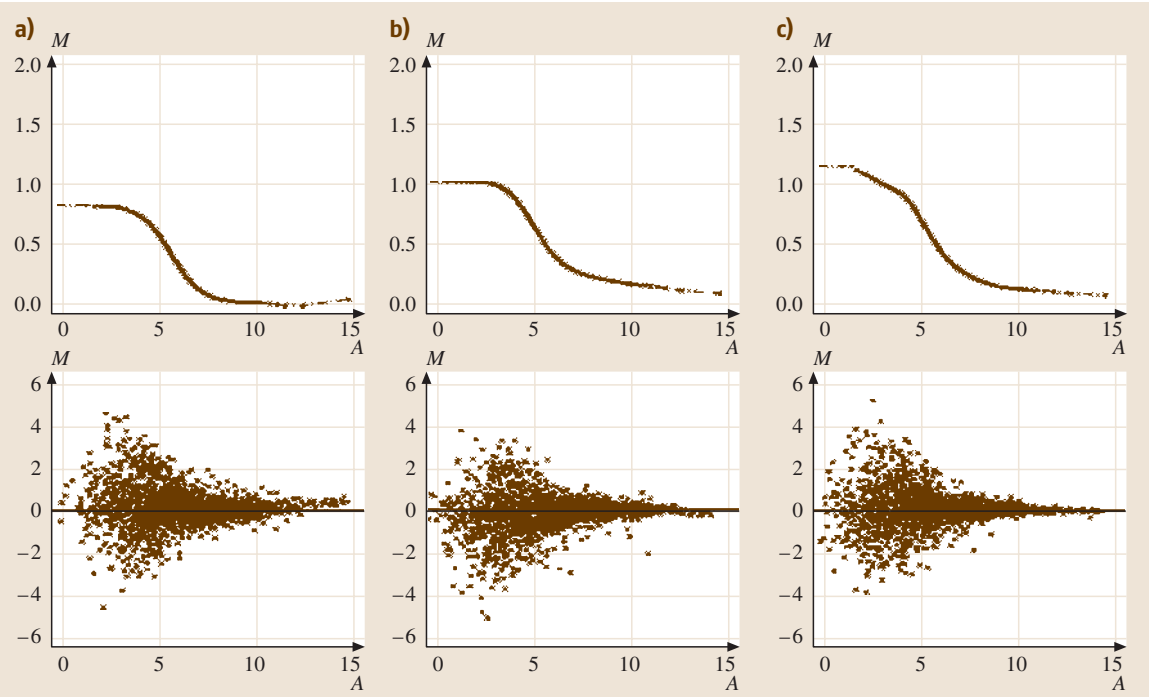


Fig. 32.1a–c Log intensity ratio $\log_2 \frac{X_{ij1}}{X_{ij2}}$ (M) as a function of average gene expression $\log_2 \sqrt{x_{ij1} x_{ij2}}$ (A). The *top row* of panels **(a)**, **(b)** and **(c)** represent local pooled error (LPE) for naive, 48 h activated, and T-cell clone D4 conditions, respectively, in the mouse immune-response microarray study in [32.13]. Variance estimates in percentile intervals are shown as *points*, and a *smoothed curve* superimposing these points is also shown. The *bottom row* of panels represent the corresponding M-versus-A graphs. The *horizontal line* represents identical expression between replicates

than ten. In a simulation study from a Gaussian distribution without extreme outliers, the LPE method showed a significant improvement of statistical power with three and five replicates (see Figure 2 in [32.13]).

32.3 Statistical Modeling for Genomic Data

Microarray gene-expression studies are also frequently performed for comparing complex, multiple biological conditions and pathways. Several linear modeling approaches have been introduced for analyzing microarray data with multiple conditions. Reference [32.23] considered an analysis of variance (ANOVA) model to capture the effects of dye, array, gene, condition, array–gene interaction, and condition–gene interaction separately on cDNA microarray data, and [32.24] proposed a two-stage mixed model that first models cDNA microarray data with the effects of array, condition, and condition–array interaction, and then fits the residuals with the effects of gene, gene–condition interaction, and gene–array interaction. Several approaches have also been developed under the Bayesian paradigm for analyzing microarray data including: the Bayesian parametric modeling [32.25], the Bayesian regularized t-test [32.21], the Bayesian hierarchical modeling with a multivariate normal prior [32.26], and the Bayesian heterogeneous error model (HEM) with two error components [32.27]. The ANOVA and HEM approaches are introduced in this chapter.

32.3.1 ANOVA Modeling

Reference [32.23] first suggested the use of analysis of variance (ANOVA) models to both estimate relative gene expression and to account for other sources of variation in microarray data. Even though the exact form of the ANOVA model depends on the particular data set, a typical ANOVA model for two-color-based cDNA microarray data can be defined as

$$y_{ijkg} = \mu + A_i + D_j + V_k + G_g + (AD)_{ij} + (AG)_{ig} + (DG)_{ig} + (VG)_{kg} + \epsilon_{ijkg} \,, \tag{32.4}$$

where y_{ijkg} is the measured intensity from array i, dye j, variety k, and gene g on an appropriate scale (typically the log scale). The generic term *variety* is often used to refer to the mRNA samples under study, such as treatment and control samples, cancer and normal cells, or time points of a biological process. The terms A, D, and AD account for all effects that are not gene-specific. The gene effects G_g capture the average levels of expression for genes and the array-by-gene interactions $(AG)_{ig}$ capture differences due to varying sizes of spots on arrays. The dye-by-gene interactions $(DG)_{jg}$ represent gene-specific dye effects. None of the above effects are of biological interest, but amount to a normalization of the data for ancillary sources of variation. The effects of primary interest are the interactions between genes and varieties, $(VG)_{kg}$. These terms capture differences from overall averages that are attributable to the specific combination of variety k and gene g. Differences among these variety-by-gene interactions provide the estimates for the relative expression of gene g in varieties 1 and 2 by

$$(VG)_{1g} - (VG)_{2g} \,.$$

Note that AV, DV, and other higher-order interaction terms are typically assumed to be negligible and considered together with the error terms. The error terms ϵ_{ijkg} are often assumed to be independent with mean zero and variance σ^2. However, such a *global* ANOVA model is difficult to implement in practice due to its computational restriction. Instead, one often considers gene-by-gene ANOVA models like:

$$y_{ijkg} = \mu_g + A_i + D_j + V_k + (AD)_{ij} + (VG)_{kg} + \epsilon_{ijkg} \,. \tag{32.5}$$

Alternatively, a two-stage ANOVA model is used [32.24]. The first layer is for the main effects that are not specific to the gene

$$y_{ijkg} = \mu + A_i + D_j + V_k + (AD)_{ij} + (AG)_{ig} + \epsilon_{ijkg} \,. \tag{32.6}$$

Let r_{ijkg} be the residuals from this first ANOVA fit. Then, the second-layer ANOVA model for gene-specific effects is considered as

$$r_{ijkg} = G_g + +(AG)_{ig} + (DG)_{ig} + (VG)_{kg} + \nu_{ijkg} \,. \tag{32.7}$$

Except the main effects of G and V and their interaction effects, the other terms A, D, (AD), (AG), and (DG) can be considered as random effects. These within-gene ANOVA models can be implemented using most standard statistical packages, such as R, SAS, or SPSS.

32.3.2 The Heterogeneous Error Model

Similarly to the statistical tests for comparing two sample conditions, the above within-gene ANOVA modeling methods are underpowered and have inaccurate error estimation in microarray data with limited replication. Using a Bayesian hierarchical approach with LPE-based (or error-pooling) empirical Bayes prior constructions, [32.27] have constructed a heterogeneous error model (HEM) with two layers of error to decompose the total error variability into the technical and biological error components in microarray data (HEM package at Bioconductor). Utilizing the LPE-estimated error-distribution information of microarray data for its empirical Bayes prior specifications, this modeling strategy provides separate estimates of the technical and biological error components in microarray data, especially the former error component, significantly more accurately. The first layer is constructed to capture the array technical variation due to many experimental error components, such as sample preparation, labeling, hybridization, and image processing

$$y_{ijkl} = x_{ijk} + \epsilon_{ijkl}\,, \quad \text{where}$$
$$\epsilon_{ijkl} \sim \text{iid Normal} \left[0, \sigma_\epsilon^2(x_{ijk})\right]\,, \tag{32.8}$$

where i = 1, 2, ..., G; j = 1, 2, ..., C; k = 1, 2, ..., m_{ij}; l = 1, 2, ..., n_{ijk}, and iid means independently and identically distributed. The second layer is then hierarchically constructed to capture the biological error component:

$$x_{ijk} = \mu + g_i + c_j + r_{ij} + b_{ijk}\,, \quad \text{where}$$
$$b_{ijk} \text{ iid Normal} \left[0, \sigma_b^2(ij)\right]\,. \tag{32.9}$$

Here, the genetic parameters are for the grand mean (shift or scaling) constant, gene, cell, interaction effects, and the biological error; the last error term varies and is heterogeneous for each combination of different genes and conditions. Note that the biological variability is individually assessed for discovery of biologically relevant expression patterns in this approach.

Bayesian posterior estimates and distributions are quite dependent on their prior specifications when the sample size is small in a microarray study. This difficulty in Bayesian applications to microarray data has been well-recognized and several authors have suggested the use of more-informative empirical Bayes priors [32.28, 29]. In these studies, empirical Bayes (EB) priors are used for defining distributions of genes with different expression patterns, e.g., distributions for equivalently and differentially expressed genes. Such specifications would be useful to determine each gene's expression pattern when the number of different expression patterns is small. However, as the number of conditions increases, the number of expression patterns increases exponentially, and these EB approaches quickly become impractical; many of these prior distributions also become unidentifiable. Conversely, the EB priors in HEM are used for specification of the two layers of error – technical and biological errors – which can be directly observed from each array data set, and can also be reliably estimated by the LPE method, pooling information from the genes with similar expression intensity. Thus, a nonparametric EB prior for the technical error $\sigma^2(x_{ijk})$ that is estimated by the LPE method and sampled by bootstrapping at each intensity x_{ijk}, whereas a parametric EB prior, Gamma (α, β_{ij}) is used because this error should freely vary to reflect the actual sampling variability of different biological subjects. Using these error-pooling-based prior specifications, HEM has demonstrated its improved performance in small sample microarray data analysis both in simulated and practical microarray data, see Fig. 32.2.

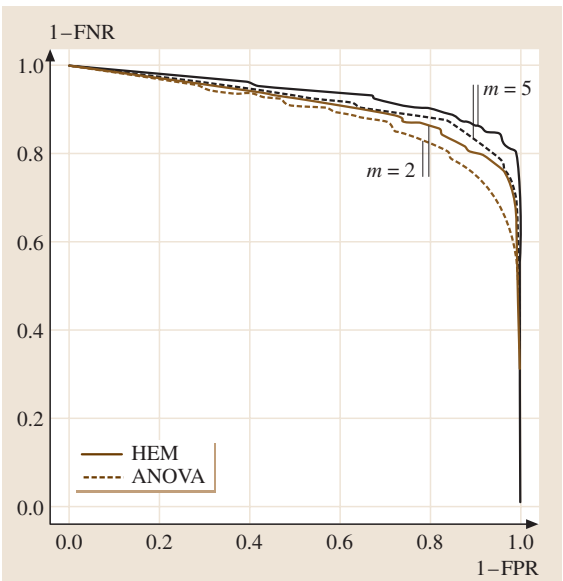


Fig. 32.2 ROC curves from HEM (*solid lines*) and ANOVA (*dotted lines*) models with two and five replicated arrays; The *horizontal axis* is 1 − FPR = 1 − Pr(positive|negative) and the *vertical axis* is 1 − FNR = 1 − Pr(negative|positive)

32.4 Unsupervised Learning: Clustering

Clustering analysis is widely applied to search for the groups (clusters) in microarray data because these techniques can effectively reduce the high-dimensional gene-expression data into a two-dimensional dendrogram organized by each gene's expression-association patterns. These clustering approaches first need to be defined by a measure or distance index of similarity or dissimilarity such as

- Euclidean: $d(x, y) = \sum_k (x_k - y_k)^2$;
- Manhattan: $d(x, y) = \sum |x_k - y_k|$;
- Correlation: $d(x, y) = 1 - r(x, y)$, where $r(x, y)$ is the Pearson or Spearman sample-correlation coefficient.

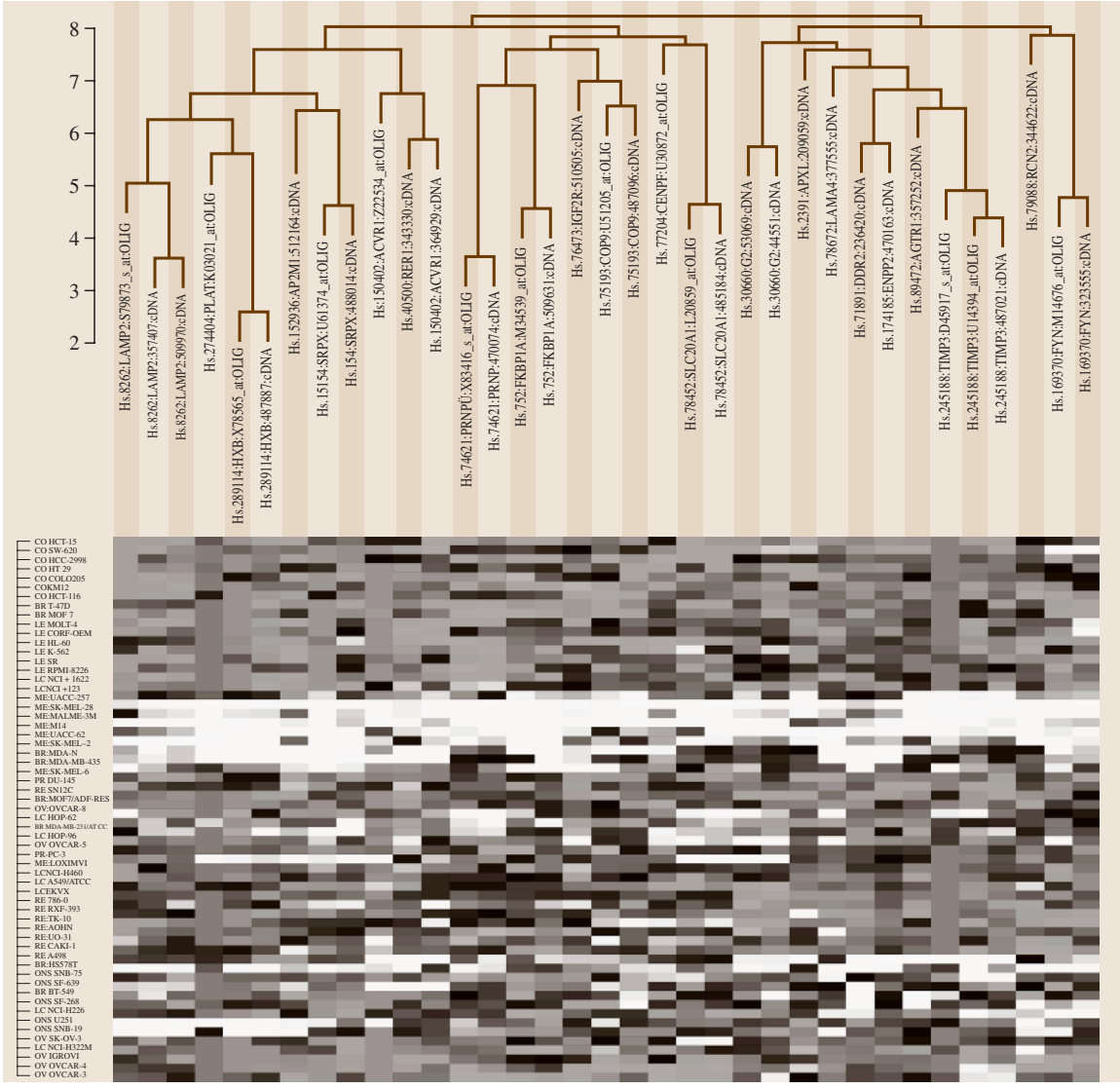


Fig. 32.3 Clustered image maps (CIMs) for hierarchical clustering of the cDNA and oligo array expression patterns. Each gene expression pattern is designated as coming from the cDNA or oligo array set. A region of CIM occupied by melanoma genes from the combined set of 3297 oligo and cDNA transcripts [32.3]

Note that if x and y are standardized, i. e., subtracted by each mean and divided by each standard deviation, then Euclidean and correlation distances can be easily shown to be mathematically equivalent:

$$\begin{aligned}\sum_k (x_k - y_k)^2 &= \sum_k \left(x_k^2 + y_k^2 - 2x_k y_k\right) \\ &= 2\left(1 - \sum_k x_k y_k\right) \\ &= 2[1 - r(x, y)] \,.\end{aligned}$$

Two classes of clustering algorithms have been used in genomic data analysis. The first class of clustering algorithms is based on hierarchical allocation including

1. Agglomerative methods:
   a) average linkage based on group average distance [32.3, 7]
   b) single linkage based on minimum nearest distance
   c) complete linkage based on maximum furthest distance.
2. Probability-based clustering: Bayes factor or posterior probability for choosing k clusters
3. Divisive methods: monothetic variable division, polythetic division

A cluster-image map is shown for the microarray data of the NCI 60 cancer cell lines in Fig. 32.3.

The second class is the partitioning algorithms that divide the data into a prespecified number of subsets including:

1. Self-organizing map: divides the data into a geometrically preset grid structure of subclusters [32.8];
2. Kmeans: iterative relocation clustering into a predefined number of subclusters;
3. Pam (partitioning around medoids) similar to, but more robust than Kmeans clustering;
4. Clara: clustering for applications to large data sets;
5. Fuzzy algorithm: provide fractions of membership, rather than deterministic allocations.

One of the most difficult aspects of using these clustering analyses is to interpret their heuristic, often unstable clustering results. To overcome this shortcoming, several refined clustering approaches have been suggested. For example, [32.23] suggest the use of bootstrapping to evaluate the consistency and confidence of each gene's membership to particular cluster groups. The *gene shaving* approach has been suggested to find the clusters directly relevant to major variance directions of an array data set [32.30]. Recently, *tight clustering*, a refined bootstrap-based hierarchical clustering is proposed to formally assess and identify the groups of genes that are most tightly clustered to each other [32.31].

32.5 Supervised Learning: Classification

Applications of microarray data have received considerable attention in many challenging classification problems in biomedical research [32.2, 32, 33]. In particular, such applications have been conducted in cancer research as alternative diagnostic techniques to the traditional ones such as classification by the origin of cancer tissues and/or the microscopic appearance; the latter are far from satisfaction for the prediction of many critical human disease subtypes [32.34]. Several different approaches to microarray classification modeling have been proposed, including gene voting [32.2], support vector machines (SVMs) [32.35, 36], Bayesian regression models [32.33], partial least squares [32.37], genetic-algorithm k-nearest-neighbor (GA/KNN) method [32.38], and between-group analysis [32.39].

Microarray data often have tens of thousands of genes on each chip whereas only a few tens of samples or replicated arrays are available in a microarray study. Therefore, it is desirable to avoid overfitting and to find a best subset of the thousands of genes for constructing classification rules and models that are robust to different choices of training samples and provide consistent prediction performance for future samples. In particular, to avoid inflated evaluation of prediction performance from a large screening search on many candidate models, feature selection must be simultaneously performed with classification model construction on a training set under a particular classification method. Evaluation of prediction performance should then be carefully conducted among the extremely large number of competing models, especially in using appropriate performance selection criteria and in utilizing the whole data for model training and evaluation.

32.5.1 Measures for Classification Model Performance

Several different measures are currently used to evaluate the performance of classification models: classification error rate, area under the receiver operating characteristics curve (AUC), and the product of posterior classification probabilities [32.40–42]. However, when a large number of candidate models, e.g., more than 10^8 two-gene models on 10 k array data, are compared in their performance, these measures are often quickly saturated; their maximum levels are achieved by many competing models, so that identification of the best prediction model among them is extremely difficult. Furthermore, these measures cannot capture an important aspect of classification model performance as follows. Suppose three samples are classified using two classification models (or rules). Suppose also that one model provides correct posterior classification probabilities 0.8, 0.9, and 0.4, and the other 0.8, 0.8, and 0.4 for the three samples. Assuming these were unbiased estimates of classification error probabilities (on future data), the former model would be preferred because this model will perform better in terms of the expected number of correctly classified samples in future data. Note that the two models provide the same misclassification error rate, 1/3. This aspect of classification performance cannot be captured by evaluating the commonly used error rate or AUC criteria, which simply add one count for each correctly classified sample ignoring its degree of classification error probability.

To overcome this limitation, the so-called *misclassification penalized posterior* (MiPP) criterion has recently been suggested [32.43]. This measure is the sum of the correct-classification (posterior) probabilities of correctly classified samples subtracted by the sum of the misclassification (posterior) probabilities of misclassified samples. Suppose there are m classes from populations π_i $(i = 1, \ldots, m)$ and a total of $N = \sum_{i=1}^{m} n_i$ samples. Let $\boldsymbol{X}_{ij}$, $j = 1, \ldots, n_i$, be the j-th sample vector from the i-th class under a particular prediction model (e.g., one-gene or two-gene model), denoted as R_{M} and a rule R, e.g., linear discriminant analysis (LDA) or SVMs. For sample vector $\boldsymbol{X}_{ij}$, the posterior classification probability to be assigned to class k (under R_{M}) is defined as $p_k(\boldsymbol{X}_{ij}) = P\left(\boldsymbol{X}_{ij} \in \pi_k | \boldsymbol{X}_{ij}\right)$. (We omit the notation R_{M} for simplicity.) For example, $p_k(\boldsymbol{X}_{kj})$ is thus the posterior probability of correct classification for the sample $\boldsymbol{X}_{kj}$. MiPP is then defined as:

$$\psi_p = \sum_{\text{correct}} p_k(\boldsymbol{X}_{kj}) - \sum_{\text{wrong}} \left[1 - p_k(\boldsymbol{X}_{kj})\right] . \tag{32.10}$$

Here correct and wrong correspond to the samples that are correctly and incorrectly classified, respectively. In the two-class problem, *correct* simply means $p_k(\boldsymbol{X}_{kj}) > 0.5$, but in general, it occurs when $p_k(\boldsymbol{X}_{kj}) = \max_{i=1,\ldots,m}[p_i(\boldsymbol{X}_{kj})]$.

It can be shown that MiPP is simply the sum of the posterior probabilities of correct classification penalized by the number of misclassified samples (N_{M})

$$\psi_p = \sum_{\text{correct}} p_k(\boldsymbol{X}_{kj}) + \sum_{\text{wrong}} p_k(\boldsymbol{X}_{kj}) - \sum_{\text{wrong}} 1 = \sum p_k(\boldsymbol{X}_{kj}) - N_{\mathrm{M}} . \tag{32.11}$$

That is, MiPP increases as the sum of correct-classification posterior probabilities increases, as the number of misclassified samples decreases, or both. Thus, MiPP is a continuous measure of classification performance that takes into account both the degree of classification certainty and the error rate, and is sensitive enough to distinguish subtle differences in prediction performance among many competing models. MiPP can be directly derived from the posterior classification probabilities of class membership in LDA, quadratic discriminant analysis (QDA), and logistic regression (LR), but it is slightly differently obtained for SVMs because they do not directly provide an estimate of posterior classification probability. In this case, a logit-link-based estimation can be used to derive a *pseudo* posterior classification probabilities as suggested by [32.44].

32.5.2 Classification Modeling

As described above, several classification modeling approaches are currently used in genomic data analysis. These approaches often adopt certain cross-validation techniques, such as leave-one-out or training-and-validation-set strategies for their modeling search and fitting.

Gene Voting

Adopting the idea of aggregating power by multiple predictors, the so-called *gene voting* classification method has been proposed for the prediction of subclasses of acute leukemia patients observed by microarray gene-expression data [32.2]. This method gains accuracy by

aggregating predictors built from a learning set and by casting their voting weights. For binary classification, each gene casts a vote for class 1 or 2 among p samples, and the votes are aggregated over genes. For gene g_j the vote is

$$v_j = a_j(g_j - b_j) ,$$

where $a_j = (\hat{\mu}_1 - \hat{\mu}_2)/(\hat{\sigma}_1 + \hat{\sigma}_2)$ and $b_j = (\hat{\mu}_1 + \hat{\mu}_2)/2$. Using this method based on 50 gene predictors, [32.2] has correctly classified 36 of 38 patients in an independent validation set between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL).

LDA and QDA

Linear discriminant analysis can be applied with leave-one-out classification as follow. Assume each of $f_k(x)$, $k = 1, \ldots, K$, follows a multivariate normal (μ_k, Σ) distribution with mean vector μ_k a common variance–covariance matrix Σ. Then,

$$\begin{aligned}
&\log Pr(G = k|X = x)/Pr(G = j|X = x) \\
&\quad = \log[f_k(x)/f_j(x)] + \log(\pi_k/\pi_j) \\
&\quad = \log(\pi_k/\pi_j) - \frac{1}{2}(\mu_k + \mu_j)^{\mathrm{T}} \Sigma^{-1}(\mu_k - \mu_j) \\
&\qquad + x^{\mathrm{T}} \Sigma^{-1}(\mu_k - \mu_j) .
\end{aligned}$$

A sample vector x_o will then be allocated to group k if the above equation is greater than zero or to group j otherwise.

The quadratic discriminant analysis can be similarly performed except that the variance–covariance matrix Σ is now considered differently for each subpopulation group. The differences between LDA and QDA are typically small, especially if polynomial factors are considered in LDA. In general, QDA requires more observations to estimate each variance–covariance matrix for each class. LDA and QDA have consistently shown high performance not because the data likely from Gaussian distributions, but more likely because simple boundaries such as linear or quadratic are sufficientto define the different classes in the data [32.42].

Logistic Regression (LR)

LR discriminant analysis requires less assumptions than the aforementioned LDA and QDA approaches. LR methods simply maximize the conditional likelihood $Pr(G = k|X)$, typically by a Newton–Raphson algorithm [32.45]. The allocation decision on a sample vector x_o by LR is based on the logit regression fit:

$$\mathrm{Logit}(p_i) = \log[p_i/(1 - p_i)] \sim \hat{\beta}^{\mathrm{T}} x ,$$

where $\hat{\beta}$ is the LR estimated coefficient vector for the microarray data. LR discriminate analysis is often used due to its flexible assumption about the underlying distribution, but if it is actually applied to a Gaussian distribution, LR shows a loss of 30% efficiency in the (misclassification) error rate compared to LDA.

Support Vector Machines (SVMs)

SVMs separate a given set of binary labeled training data with a hyperplane that is maximally distant from them; this is known as the maximal margin hyperplane [32.35]. Based on a kernel, such as a polynomial of dot products, the current data space will be embedded in a higher-dimensional space. The commonly used kernels are:

- Radial basis kernel: $K(x, y) = \exp\left(-\frac{|x-y|^2}{2\sigma^2}\right)$,
- Polynomial kernel: $K(x, y) = < x, y >^d$ or $K(x, y) = (< x, y > +c)^d$,

where $<$, $>$ denotes the inner-product operation. Note that the above polynomial kernel is of order d and is linear when $d = 1$. Using a training set, we derive a hyperplane with maximal separation and validate against a validation set.

SVMs often consider linear classifiers:

$$f_{w,b}(x) = < w, x > +b ,$$

which lead to linear prediction rules: $h_{w,b}(x) = \mathrm{sign}[f_{w,b}(x)]$ for the decision boundary of the hyperplane $f_{w,b}(x)$. SVMs maps each vector-valued example into a feature space:

$$x \to [\psi_1(x), \psi_2(x), \ldots, \psi_N(x)] .$$

32.5.3 Stepwise Cross-Validated Discriminant Analysis

Classification techniques must be carefully applied in prediction model training on genomic data. In particular, if all the samples are used both for model search/training and for model evaluation in a large screening search for classification models, a serious selection bias is inevitably introduced [32.46]. In order to avoid such a pitfall, a stepwise (leave-one-out) cross-validated discriminant procedure (SCVD) that gradually adds genes to the training set has been suggested [32.42, 47]. It is typically found that the prediction performance is continuously improved (or not decreased) by adding more features into the model. This is again due to a sequential search and selection

strategy against an astronomically large number of candidate models; some of them can show over-optimistic prediction performance for a particular training set by chance. Note also that even though a leave-one-out or similar cross-validation strategy is used in this search, the number of candidate models is too big to eliminate many random ones that survived from such a specific cross-validation strategy by chance. Thus, test data should be completely independent from the training data to obtain an unbiased estimate of each model's performance.

The SCVD Procedure

Using the MiPP criterion above, the SCVD classification model is constructed sequentially as follows. Given a classification rule R, the models are constructed on a training data set in a forward stepwise cross-validated discriminant fashion. Suppose we have a training data set consisting of N samples and g candidate features (genes). A schematic summary of the MiPP-based SCVD model construction is shown in Fig. 32.4.

The initial step begins by fitting each feature individually on the training set. For each of the G features, MiPP is calculated based on leave-one-out cross-validation (so MiPP for a gene is the average of MiPPs of the N leave-one-out fits for that particular gene). The gene with the maximal value of MiPP is then retained, and the optimal one-gene model O_1 is fit using all training samples. The second step adds each of the $G-1$ features and of these $G-1$ two-gene models, the one with the maximal value of MiPP is similarly retained and used to construct the optimal two-gene model O_2. This process continues adding features in this sequential fashion until the training model becomes saturated at the L-th step, i. e., MiPP converges to a certain maximum level and the L-gene MiPP is not bigger than the (L-1)-gene MiPP (note that MiPP has an upper bound of N).

Because of the sequential selection of features in this model construction, the performance of the prediction model improves when there a large number of features in a model and this cannot be used as an objective measure of classification performance. Therefore, the performance of each of the optimal models $O_1, \ldots, O_L$ is assessed on a completely independent test data set to determine the final robust prediction model. In this case, both MiPP and the error rate can be evaluated since the latter can be used among the small number of competing optimal models with different numbers of model features.

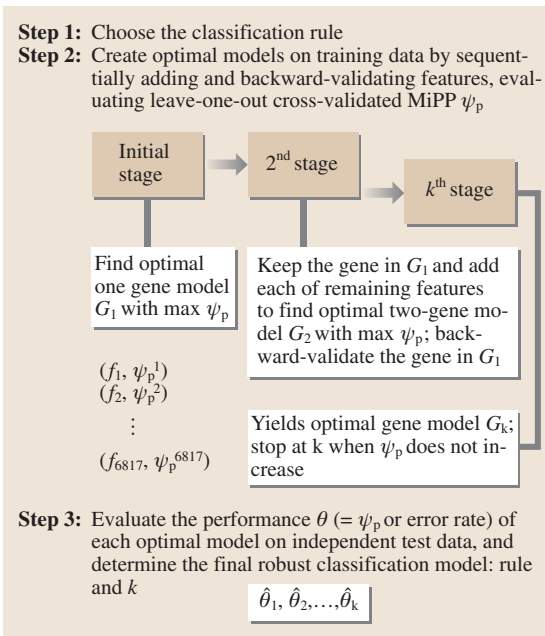


Fig. 32.4 A schematic diagram for classification modeling based on the stepwise cross-validated discriminant (SCVD) procedure

Comparison of Classification Methods

Using this SCVD strategy based on MiPP, several widely used classification rules such as linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logistic regression (LR), and support vector machines

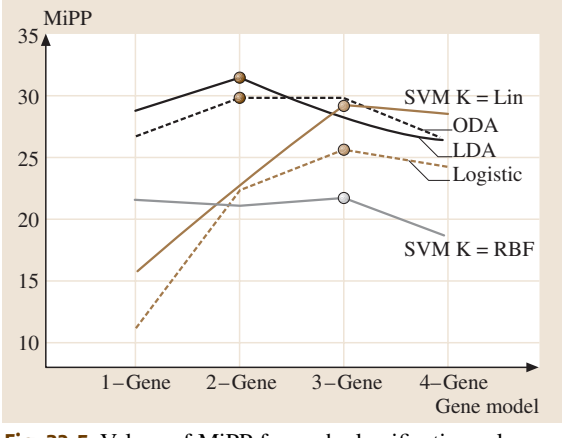


Fig. 32.5 Values of MiPP for each classification rule constructed for models with up to four genes. The best gene model of all the gene models for a given classification rule is denoted by a •

Table 32.2 Classification results of the classification rules and the corresponding gene model

Method	Model	Training data ER%	ψ_p	Test data ER %	ψ_p
LDA	1882+1144	0	37.91	2.94	31.46
QDA	4847+5062	0	37.96	5.88	29.81
Logistic	1807+4211+575	0	37.998	11.76	25.64
SVM K=Linear	2020+4377+1882	0	35.16	0	29.26
SVM K=RBF	4847+3867+6281	0	32.52	5.88	21.713

(SVMs) with linear or radial basis function (RBF) kernels have been compared. The leukemia microarray data in [32.2] had a training set of 27 ALL and 11 AML samples and an independent test set of 20 ALL and 14 AML samples. Since two distinct data sets exist, the model is constructed on the training data and evaluated on the test data set. Figure 32.5 shows the performance of each classification rule on the test data set. Each rule identified a different subset of features and the performance of the best subset for each classification rule along with its performance is shown in Table 32.2. This best subset is simply the point at which each line from Fig. 32.5 reaches its maximum value.

In terms of error rate, it appears as if the SVM with a linear kernel is the most accurate rule. However, LDA only misclassified one sample and the SVM with the RBF kernel and QDA misclassified two samples on the independent test data. Logistic regression does not seem to perform as well as the other rules, by misclassifying 4 out of 34 samples. Note again that comparing the rules on the basis of MiPP is somewhat tricky for SVMs since the estimated probabilities of correct classification from SVMs are based upon how far samples are from a decision boundary. As a result, unlike the LDA, QDA, and LR cases, these are not true posterior classification probabilities. In an application to a different microarray study on colon cancer, the RBF-kernel SVM model with three genes was found to perform best among these classification techniques.

Therefore, using the MiPP-based SCVS procedure, the most parsimonious classification models were derived with a very small number of features, only two or three genes from microarray data, outperforming many previous models with 50–100 features. This may imply that a set of a small number of genes may be sufficient to explain the discriminativeinformation of different types of a particular desease, even though it is often found that there exist multiple sets (of small numbers of genes) with similar classification prediction performance.

References

32.1 C. Sander: Genomic medicine and the future of health care, **287**, 1977–8 (2000)

32.2 T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, E. S. Lander: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science **286**, 5439 (1999)

32.3 J. K. Lee, U. Scherf, K. J. Bussey, F. G. Gwadry, W. Reinhold, G. Riddick, S. L. Pelletier, S. Nishizuka, G. Szakacs, J.-P. Annereau, U. Shankavaram, S. Lababidi, L. H. Smith, M. M. Gottesman, J. N. Weinstein: Comparing cDNA, oligonucleotide array data: Concordance of gene expression across platforms for the NCI-60 cancer cell lines, Genome Biol. **4**, R82 (2003)

32.4 D. Pinkel: Cancer cells, chemotherapy, gene clusters, Nat. Genet. **24**, 208–9 (2000)

32.5 J. K. Lee: Discovery, validation of microarray gene expression patterns, LabMedica Int. **19**, 8–10 (2002)

32.6 C. J. Stoeckert, H. C. Causton, C. A. Ball: Microarray databases: standards, ontologies, Nat. Genet. **32**, 469–473 (2002)

32.7 M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein: Cluster analysis, display of genome-wide expression patterns, Proc. Nat. Acad. Sci. **95**, 14863–8 (1998)

32.8 P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, T. R. Golub: Interpreting patterns of gene expression with self-organizing maps: Methods, application to hematopoietic differentiation, Proc. Nath. Acad. Sci. **96**, 2907–2912 (1999)

32.9 S. Dudoit, Y. H. Yang, M. J. Callow, T. P. Speed: Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments, Stat. Sin. **12**, 111–139 (2002)

32.10 V. Tusher, R. Tibshirani, C. Chu: Significance analysis of microarrays applied to transcriptional responses to ionizing radiation, Proc. Nat. Acad. Sci. **98**, 5116–21 (2001)

32.11 Y. Benjamini, Y. Hochberg: Controlling the false discovery rate: a practical, powerful approach to multiple testing, J. R. Stat. Soc., Ser. B, Methodological **57**, 289–300 (1995)

32.12 J. Storey, R. Tibshirani: SAM thresholding, false discovery rates for detecting differential gene expression in DNA microarrays. In: *The Analysis of Gene Expression Data: Methods and Software*, ed. by G. Parmigiani, E. S. Garrett, R. A. Irizarry, S. L. Zeger (Springer, Berlin Heidelberg New York 2003) Chap. 12

32.13 N. Jain, K. Ley, J. Thatte, M. O'Connell, J. K. Lee: Local pooled error test for identifying differentially expressed genes with asmall number of replicated microarrays, Bioinformatics **19**, 1945–51 (2003)

32.14 W. Jin, R. M. Riley, R. D. Wolfinger, K. P. White, G. Passador-Gurgel, G. Gibson: The contributions of sex, genotype, age to transcriptional variance in Drosophila melanogaster, Nat. Genet. **29**, 389–395 (2001)

32.15 A. Kamb, A. Ramaswami: A simple method for statistical analysis of intensity differences in microarray-derived gene expression data, BMC Biotechnol. **1**, 1–8 (2001)

32.16 R. Nadon, P. Shi, A. Skandalis, E. Woody, H. Hubschle, E. Susko, P. Ramm, N. Rghei: Statistical inference methods for gene expression arrays, BIOS **4266**, 46–55 (2001)

32.17 B. Durbin, J. Hardin, D. Hawkins, D. Rocke: A variance-stabilizing transformation for gene-expression microarray data, Bioinformatics **18**, 1105 (2002)

32.18 X. Huang, W. Pan: Comparing three methods for variance estimation with duplicated high density oligonucleotide arrays, Funct. Integr. Genomics **2**, 126–133 (2002)

32.19 Y. Lin, S. T. Nadler, A. D. Attie, B. S. Yandell: Adaptive gene picking with microarray data: detecting important low abundance signals. In: *The Analysis of Gene Expression Data: Methods and Software*, ed. by G. Parmigiani, E. S. Garrett, R. A. Irizarry, S. L. Zeger (Springer, Berlin Heidelberg New York 2003) Chap. 13 (http://www.stat.wisc.edu/~yilin/)

32.20 I. Lönnstedt, T. P. Speed: Replicated microarray data, Stat. Sin. **12**, 31–46 (2002)

32.21 P. Baldi, A. D. Long: A Bayesian framework for the analysis of microarray expression data: regularized t-test, statistical inferences of gene changes, Bioinformatics **17**, 509–519 (2001)

32.22 J. K. Lee, M. O'Connell: An S-PLUS library for the analysis of differential expression. In: *The Analysis of Gene Expression Data: Methods and Software*, ed. by G. Parmigiani, E. S. Garrett, R. A. Irizarry, S. L. Zeger (Springer, Berlin Heidelberg New York 2003) Chap. 7

32.23 M. K. Kerr, G. A. Churchill: Statistical design, the analysis of gene expression microarray data, Genetic Res. **77**, 123–128 (2001)

32.24 R. D. Wolfinger, G. Gibson, E. D. Wolfinger, L. Bennett, H. Hamadeh, P. Bushel, C. Afshari, R. S. Pales: Assessing gene significance from cDNA microarray expression data via mixed models, J. Comput. Biol. **8**, 37–52 (2001)

32.25 M. A. Newton, C. M. Kendziorski, C. S. Richmond, F. R. Blattner, K. W. Tsui: On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data, J. Comp. Biol. **8**, 37–52 (2001)

32.26 J. G. Ibrahim, M.-H. Chen, R. J. Gray: Bayesian models for gene expression with DNA microarray data, J. Am. Stat. Assoc. **97**, 88–99 (2002)

32.27 H. J. Cho, J. K. Lee: Hierarchical error model for analyzing gene expression data, Bioinformatics **20**, 2016–2025 (2004)

32.28 B. Efron, R. Tibshirani, J. D. Storey, V. Tusher: Empirical bayes analysis of a microarray experiment, J. Am. Stat. Assoc. **96**, 1151–1160 (2001)

32.29 M. A. Newton, C. K. Kendziorski: Parametric empirical bayes methods for microarrays. In: *The Analysis of Gene Expression Data: Methods and Software*, ed. by G. Parmigiani, E. S. Garrett, R. A. Irizarry, S. L. Zeger (Springer, Berlin Heidelberg New York 2003)

32.30 T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, P. Brown: 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns, Genome Biol. **1**, Research03 (2000)

32.31 G. C. Tseng, W. H. Wong: Tight clustering: a resampling-based approach for identifying stable and tight patterns in data, Biometrics **61**(1), 10–16 (2004)

32.32 U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, A. J. Levine: Broad patterns of gene expression revealed by clustering analysis of tumor, normal colon tissues probed by oligonucleotide arrays, Proc. Nath. Acid. Sci. **96**, 6745–6750 (1999)

32.33 M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. R. Marks, J. R. Nevins: Prediction the clinical status of human breast cancer by using gene expression profiles, Proc. Nath. Acad. Sci. **98**, 11462–11467 (2001)

32.34 J. Staunton, D. Slonim, P. Tanamo, M. Angelo, J. Park, U. Scherf, J. K. Lee, W. Reinhold, J. Weinstein, J. Mesirov, E. Lander, T. Golub: Chemosensitivity prediction by transcriptional profiling, Proc. Natl. Acad. Sci **11;98**(19), 10787–10792 (2001)

32.35 T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, D. Haussler: Support vector ma-

chine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics **16**, 906–914 (2000)

32.36 S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J. P. Mesirov, T. Poggio: *Support Vector Machine Classification of Microarray Data* (MIT, Cambridge 1998)

32.37 D. V. Nguyen, D. M. Rocke: Tumor classification by partial least squares using microarray gene expression data, Bioinformatics **18**, 39–50 (2002)

32.38 L. Li, C. R. Weinberg, T. A. Darden, L. G. Pedersen: Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics **17**, 1131–1142 (2001)

32.39 A. C. Culhane, G. Perriere, E. C. Considine, T. G. Cotter, D. G. Higgins: Between-group analysis of microarray data, Bioinformatics **18**, 1600–1608 (2002)

32.40 A. P. Bradley: The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recog. **30**, 1145–1159 (1997)

32.41 D. J. Hand: *Construction and Assessment of Classification Rules* (Wiley, Chichester 1997)

32.42 M. Soukup, J. K. Lee: Developing optimal prediction models for cancer classification using gene expression data, J. Bioinf. Comp. Biol. **1**, 681–694 (2004)

32.43 M. Soukup: Robust optimization of classification model for predicting human disease subtypes using microarray gene expression data. Ph.D. Thesis (University of Virginia, Charlottesville 2004)

32.44 G. Wahba: Support vector machines, reporoducing Kenel Hilbert spaces, the randomized GACV. In: *Advances in Kernel Methods-Support Vector Learning*, ed. by B. Scholkopf, C. J. C. Burges, A. J. Smola (MIT Press, Cambridge 1999) pp. 69–88

32.45 F. C. Pampel: *Logistic Regression: A Primer.*, Sage Univ. Papers Ser. Quant. Appl. Social Sci. (Thousand Oaks, Sage 2000) pp. 07–132

32.46 C. Ambroise, G. J. McLachlan: Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Nath. Acid. Sci. **10**, 6562–6566 (2002)

32.47 M. Soukup, H. Cho, J. K. Lee: Robust classification modeling on microarary data using misclassification penalized posterior, Bioinformatics **21**(1), i423–i430 (2005)

33. Statistical Methodologies for Analyzing Genomic Data

The purpose of this chapter is to describe and review a variety of statistical issues and methods related to the analysis of microarray data. In the first section, after a brief introduction of the DNA microarray technology in biochemical and genetic research, we provide an overview of four levels of statistical analyses. The subsequent sections present the methods and algorithms in detail.

In the second section, we describe the methods for identifying significantly differentially expressed genes in different groups. The methods include fold change, different *t*-statistics, empirical Bayesian approach and significance analysis of microarrays (SAM). We further illustrate SAM using a publicly available colon-cancer dataset as an example. We also discuss multiple comparison issues and the use of false discovery rate.

In the third section, we present various algorithms and approaches for studying the relationship among genes, particularly clustering and classification. In clustering analysis, we discuss hierarchical clustering, *k*-means and probabilistic model-based clustering in detail with examples. We also describe the adjusted Rand index as a measure of agreement between different clustering methods. In classification analysis, we first define some basic concepts related to classification. Then we describe four commonly used classification methods including linear discriminant analysis (LDA), support vector machines (SVM), neural network and tree-and-forest-based classification. Examples are included to illustrate SVM and tree-and-forest-based classification.

The fourth section is a brief description of the meta-analysis of microarray data in three different settings: meta-analysis of the same biomolecule and same platform microarray data, meta-analysis of the same biomolecule but different platform microarray data, and meta-analysis of different biomolecule microarray data.

We end this chapter with final remarks on future prospects of microarray data analysis.

Since the seminal work on microarray technology of *Schena* et al. [33.1], microarray data have attracted a great deal of attention, as reflected by the ever increasing number of publications on this technology in the past decade. The applications of the microarray technology encompass many fields of science from the search for differentially expressed genes [33.2], to the understanding of regulatory networks [33.3], DNA sequencing and mutation study [33.4], single nucleotide polymorphism (SNP) detection [33.5], cancer diagnosis [33.6], and drug discovery [33.7].

Accompanying the advancement of the microarray technology, analyzing microarray data has arguably become the most active research area of statistics and bioinformatics. Figure 33.1 provides a four-level overview of the analytic process. The first challenge in dealing with the microarray data is to preprocess the data, which involves background subtraction, array normalization, and probe-level data summarization. The purpose of this preprocessing is to remove noise and artifacts in order to enhance and extract hybridization signals. This data preprocessing is also often referred as

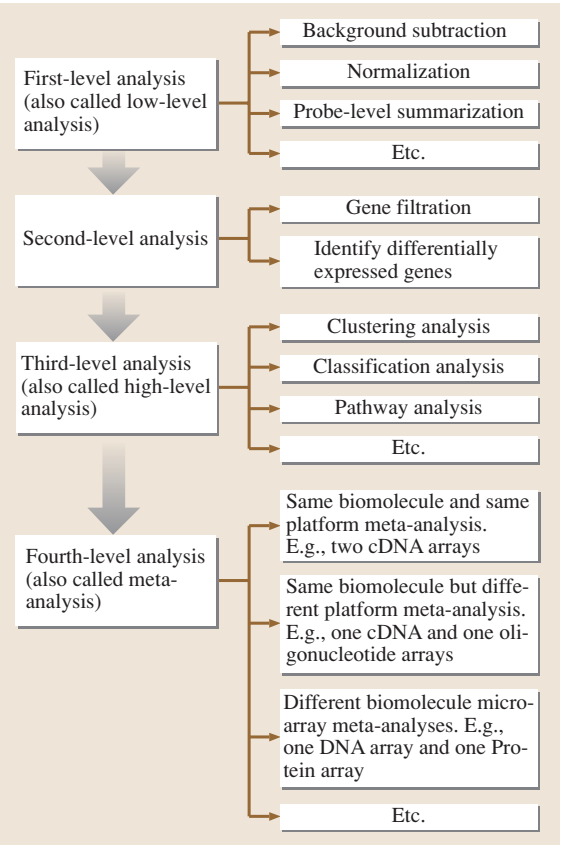


Fig. 33.1 Diagram of the four-level analysis of microarray data

the low-level analysis [33.8]. After the data are processed and cleaned, they are analyzed for different purposes. The focus of this article is on the methods for this postprocessing analysis.

The second-level analysis usually contains two steps: one is to filter *unusual* genes whose expression profiles are suspicious due to noise or are too extreme, and the other is to identify the differentially expressed genes across different samples. The gene filtration process is generally heuristic and specific to known biological contents. Thus, we will not discuss it here. To identify genes that have significantly different expression profiles, the commonly used approaches include the estimation of fold change, Student's T-test, the Wilcoxon rank sum test, the penalized T-test, empirical Bayes [33.9], and significance analysis of microarray (SAM, *Tusher* et al. [33.10]). We will review these methods in Sect. 33.1.

We will review the third-level analysis in Sect. 33.2. This type of analysis is also called high-level analysis [33.11], and it includes clustering, classification and pathway analysis. This is usually conducted on a subset of genes that are selected from the second-level analysis. To identify genes that may be correlated to each other, clustering analysis has become particularly popular, and the approaches include hierarchical clustering [33.12], k-means [33.13], self-organization maps (SOM) [33.14], principle-component analysis (PCA) [33.15], and probabilistic model-based clustering [33.16].

To classify tissue samples or diagnose diseases based on gene expression profiles, both classic discriminant analysis and contemporary classification methods have been used and developed. The methods include k-nearest neighbors (KNN) [33.17], linear discriminant analysis (LDA) [33.18], support vector machine (SVM) [33.19], artificial neural networks (ANN) [33.20], classification trees [33.21], and random and deterministic forests [33.18]. It is noteworthy that tree- and forest-based approaches can be easily applied to the entire microarray dataset without restricting our attention to a subset of selected genes.

To identify genes that may be on the same pathway of a particular biological process, relevance networks [33.22], linear differential equation [33.23], Boolean networks [33.24], Bayesian networks [33.25] and the probabilistic rational model (PRM) [33.26] have been used and developed.

The fourth-level analysis, also referred as meta-analysis, is a relative new topic for the analysis of microarray data. Because many different types and platforms of microarrays can be designed to address the same (or similar) biological problems, it is useful to compare and synthesize the results from different studies.

Before we introduce specific methods, we should point out that, as a result of high-throughput technology, the unique challenge from analyzing microarray data is the large number of genes (tens of thousands) and relatively small sample sizes (commonly on the order of tens or hundreds). In this article, n denotes the number of genes and m the number of arrays. n is generally much greater than m.

33.1 Second-Level Analysis of Microarray Data

33.1.1 Notation

For a two-channel cDNA microarray data [33.1], we have a $2n \times m$ matrix of imaging data reflecting the red (cy5) and green (cy3) signals for each of the n genes on m arrays. The log ratio of the red to green signal is usually taken for each gene, and the analysis will be based on an $n \times m$ data matrix.

For one-channel Affymetrix Oligonucleotide GeneChip data [33.27], we have a $2\sum_{i=1}^{n} p_i \times m$ matrix of raw image data where p_i is the number of probes for the i-th gene. Note that, for each probeset, Affymetrix uses a pair of perfect match (PM) and mismatch (MM). As for oligonucleotide microarrays, steps [differences, ratios, analysis of variance (ANOVA) models, etc.] can be taken to summarize the PM and MM signals for each gene, and we still have an $n \times m$ data matrix.

A major objective of microarray analysis is to infer significantly differentially expressed genes (abbreviated as SDE genes) across different samples, e.g., m_1 tumor samples versus m_2 normal samples.

Let $Y_{ij,k}$ be the expression level of the i-th gene on the j-th array in the k-th sample. Let $\overline{Y_{i.,1}}$ and $\overline{Y_{i.,2}}$ denote the average expression level of the i-th gene in samples 1 and 2, respectively.

33.1.2 Fold Change

Many studies identify SDE genes in two samples based on simple fold-change thresholds such as a two-fold change in means. Although the choice of a threshold is somewhat arbitrary, fold change is intuitive and biologically meaningful, and serves as an effective preliminary step to eliminate a large portion of genes whose data are of little interest in a particular study.

33.1.3 *t*-Statistic

As in many clinical studies, the t-statistic provides a simple, extremely useful tool to compare the data from two samples. Let $\overline{M}$ be the mean difference between the expression profiles of a gene in two groups and $\mathrm{se}(\overline{M})$ be the standard error of $\overline{M}$. The t-statistic, defined as

$$t = \frac{\overline{M}}{\mathrm{sd}(\overline{M})} ,$$

is useful to test a null hypothesis that the gene is not differentially expressed in the two groups against the alternative hypothesis that the gene is differentially expressed.

Unlike a typical clinical study, in which we have one pair or a very few pairs of hypotheses to test, in microarray analysis we have a pair of hypotheses for every gene of interest. This means that we inevitably deal with the multiple comparison issue. Although this issue is difficult and there is no clear-cut, ideal answer, many reasonable solutions have been proposed.

Efron et al. [33.9] proposed to inflate $\mathrm{se}(\overline{M})$ by adding a constant that equals the 90-th percentile of the standard errors of all the genes. *Tusher* et al. [33.10] call such a constant a fudge factor, and propose to estimate it by minimizing the coefficient of variation of the absolute t-values. We will discuss this approach in detail in Sect. 33.1.4. Other approaches have also been proposed; for example, *Smyth* [33.28] replaces $\mathrm{se}(\overline{M})$ with a Bayesian shrinkage estimator of the standard deviation.

The permutation test is also commonly used to compare the microarrays. Permutations are usually performed at the array level to create a situation similar to the null hypothesis while maintaining the dependence structure among the genes [33.10]. In every permutation, a t-statistic can be calculated for each gene. Once a large number of permutations are completed, we have an empirical distribution for the t-statistic under the null hypothesis, which then can be used to identify SDE genes.

33.1.4 The Multiple Comparison Issue

As we mentioned earlier, we have to control the type I error rate α while testing a large number of hypotheses simultaneously. There are two commonly used approaches to deal with this issue. One is to control the family-wise error rate (FWER) and the other is to control the false discovery rate (FDR).

The FWER controls the probability of making at least one false positive call at the desired significance level. FWER guarantees that the type I error rate is less than or equal to a specified value for any given set of genes. The most known example of FWER is Bonferroni correction that divides the desired significance level α by total number of hypotheses. If the desired significance level is 0.05 and we compare expression profiles in 10 000 genes, a gene is declared to have significantly different profiles in two groups if the P-value is not greater than $\frac{0.05}{10\,000} = 5 \times 10^{-6}$. Another FWER approach is the so-called *Šidák* correction in which the

adjusted type I error rate is at $1-(1-\alpha)^{\frac{1}{n}}$ [33.29], which is close to α/n. Clearly, Bonferroni and Šidák corrections are sufficient but not necessary conditions [33.30], and FWER approaches are generally very conservative and set a stringent bar to declare SDE genes.

Because of the conservative nature of the FWER approaches, the FDR concept has flourished since it was proposed by [33.31]). FDR is defined as the mean of the ratio of the number, denoted by V, of falsely rejected hypotheses to the total rejected hypotheses, denoted by R, namely,

$$\mathrm{FDR} = E\left(\frac{V}{R}|R>0\right)\Pr(R>0)\,,$$

where $\Pr(R>0)$ is the probability of rejecting at least one hypothesis.

The FDR can be controlled at a given α level through the following steps. First, for n genes, we have n null hypotheses and n p values, denoted by $p_1,\ldots,p_n$. Then, we sort the p-values in ascending order such that $p_{(1)}\le\cdots\le p_{(n)}$. We reject any gene i that satisfies the condition $p_{(i)}\le\frac{i}{n}\times\frac{\alpha}{p_0}$, where p_0 is the proportion of genes for which the null hypotheses are indeed true. Because p_0 is unknown in practice, the most conservative approach is to replace it with 1. Recently, attempts have been made to estimate p_0 as in Tusher et al.'s SAM, where they used a permutation procedure to estimate p_0. Similar to the classical p-values, the significance measures for each gene in terms of FDR are called q-values, a name that was introduced by *Storey* [33.32, 33].

In addition, the FDR concept has been generalized. For example, *Storey* and *Tibshirani* [33.9] and *Storey* et al. [33.32] proposed positive FDR (pFDR), which corrects the error rate only when they are positive findings. For microarray data, many gene profiles are correlated, *Troendle* [33.34] proposed an adjusted FDR to address the correlation and demonstrated the benefit in terms of gained power.

33.1.5 Empirical Bayesian Approach

Using microarray data from a breast cancer study, *Efron* et al. [33.9, 35] described the empirical Bayesian method. As an initial step, a summary statistic, Z, needs to be defined for every gene to reflect the scientific interest; this can be the t-statistic as described above, a Wilcoxon rank statistic, or another choice. All genes are perceived to belong to either the differentially or nondifferentially expressed group. The density of Z_i is $f_0(z_i)$ if gene i is in the nondifferentially expressed group, and $f_1(z_i)$ otherwise. Without knowing the group, Z_i has the following mixture distribution:

$$p_0 f_0(z_i)+p_1 f_1(z_i)\,,$$

where p_0 is the prior probability that gene i is not differentially expressed, and $p_1=1-p_0$.

Based on Bayes' theorem, the posterior probability that gene i is not differentially expressed given Z_i is

$$p_0(z_i)=p_0\frac{f_0(z_i)}{f(z_i)}\,.$$

We can estimate the mixture density $f(z_i)$ by the empirical distribution $\hat{f}(z_i)$ because the genes of interest are naturally a mixture of the two groups. In addition, the null density $f_0(z_i)$ can be estimated through the permutation that artificially generates data under the null hypothesis. In other words, we can derive the posterior probability $p_0(z_i)$ for a given prior p_0. The choice of p_0 can be subjective. One conservative possibility is to choose p_0 to be the minimum of $\hat{f}(z_i)/\hat{f}_0(z_i)$ so that the posterior probability $p_1(z_i)$ that gene i is differentially expressed is non-negative. Note that $p_1(z_i)=1-p_0(z_i)$. Finally, all genes can be ranked according to $p_1(z_i)$ and highly probably differentially expressed genes can be selected.

Efron et al. [33.9, 35] did not assume a specific form for $f(z_i)$. In contrast, *Lonnstedt* and *Speed* [33.36] assumed that the data comes from the mixture of normal distributions and used the conjugate priors for the variances and the means. Under those assumptions, they derived the log odds posterior test. *Smyth* [33.28] extended the hierarchical model of *Lonnstedt* and *Speed* [33.36] to deal with microarray experiments with more than two sample groups. The method is called the Limma algorithm.

33.1.6 Significance Analysis of Microarray (SAM)

Tusher et al. [33.10] introduced the SAM algorithm. SAM identifies genes with statistically significant changes in expression by assimilating a set of gene-specific t-tests in which the standard error is adjusted by adding a small positive constant. It performs a random permutation among experiments and declares the significant genes based on a selected threshold. For the given threshold, SAM estimates the FDR by comparing the number of genes significant in the permuted samples with the number of genes significant in the original sample.

SAM can be downloaded from http://www-stat.stanford.edu/~tibs/SAM/. Specifically, first, for each gene i, SAM computes a t-like statistic

$$t_i = \frac{r_i}{s_i + s_0} ,$$

where r_i is the difference between the expression means of gene i in the two groups (expression is on a logarithm scale), s_i is the standard error, and s_0 is the fudge factor to be estimated. Secondly, similarly to the FDR scheme, all t_i values are sorted into the order statistics

$$t_{(1)} \le t_{(2)} \le \ldots \le t_{(n)} .$$

To choose the significance threshold, the expression data are permuted in the two groups within each gene B times, and during each permutation, we repeat the first two steps, which leads to a set of order statistics:

$$t^b_{(1)} \le t^b_{(2)} \le \ldots \le t^b_{(n)} .$$

After the permutations, we calculate the mean of the order statistics for each gene as follows

$$\bar{t}_{(i)} = \frac{1}{B} \sum_{b=1}^{B} t^b_{(i)} .$$

For a given threshold Δ, a gene is considered significant if $|t_{(i)} - \bar{t}_{(i)}| > \Delta$, and the FDR is estimated by the ratio of the number of genes found to be significant in the permutation samples to the number of genes called significant in the original sample.

Example 1: Identification of SDE Genes Using SAM. In this example, we apply SAM to examine a publicly available colon-cancer dataset [33.37]. This dataset contains the expression profiles of 2000 genes using an Affymetrix oligonucleotide array in 22 normal and 40 colon-cancer tissues.

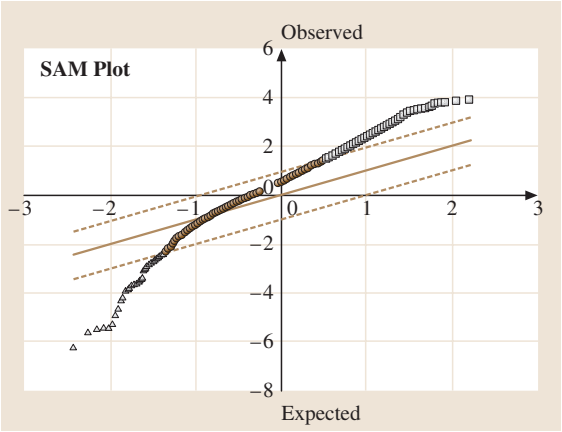


Fig. 33.2 The quantile–quantile plot from SAM for the colon-cancer dataset. Genes are declared significantly changed when their corresponding t-values are outside the two dashed lines. The white square and triangle points correspond to the genes that are significantly overexpressed and underexpressed, respectively

Figure 33.2 displays the quantile–quantile plot from SAM. The two dashed lines determine a boundary to call genes SDE depending on the choice of Δ. For example, Δ was chosen as 0.9857 in Fig. 33.2 to control the FDR at about 5%. The white square and triangle points in the figure correspond to the genes that are declared to be significantly overexpressed and underexpressed respectively. Out of the 490 declared SDE genes (440 overexpressed and 50 underexpressed), 25 genes are expected to be declared falsely.

33.2 Third-Level Analysis of Microarray Data

The third-level microarray analysis includes clustering, classification and pathway analysis. These approaches usually, though not always, follow the second-level microarray analysis because most of them can work effectively on only a small number of genes.

33.2.1 Clustering

Clustering is arguably the most commonly used approach at the third-level of analysis [33.38, 39]. It is an unsupervised learning algorithm from a machine-learning viewpoint, because the gene classes are unknown or not used, and need to be *discovered* from the data. Therefore, the goal of clustering analysis is to group genes (or arrays) based on their similarity in the feature space (e.g., expression pattern).

The underlying assumption behind clustering is that genes with similar expression profiles should share some common biological behaviors, e.g., belonging to the same protein complex or gene family [33.40], having common biological functions [33.41], being regulated by common transcription factors [33.3], belonging to the same genetic pathway, or coming from the same origin [33.39].

After the clusters are formed, a dendrogram or a tree of all genes will be viewed, although the views are not unique, because there is a left-or-right selection at each splitting step. Two popular programs for gene clustering are *Eisen* et al.'s TreeView program [33.12] and Li and Wong's dChip programs [33.8]. Routines are also available in standard statistical packages such as R, Splus, and SAS.

Distance

In order to group objects (genes or arrays) together, we need to define a measure to quantify the similarity among objects in the feature space. Such a measure of similarity is called a distance. There are several commonly used definitions of distance. Suppose that the expression profiles of two genes are $Y_i = (y_{i1}, y_{i2}, \ldots, y_{im})$ and $Y_j = (y_{j1}, y_{j2}, \ldots, y_{jm})$.

The Euclidean distance between Y_i and Y_j is

$$d_{\mathrm{E}}\left(Y_i, Y_j\right) = \left[\sum_{k=1}^{m}\left(y_{ik} - y_{jk}\right)^2\right]^{\frac{1}{2}} .$$

The city-block distance between Y_i and Y_j is

$$d_{\mathrm{C}}\left(Y_i, Y_j\right) = \sum_{k=1}^{m}\left|y_{i1} - y_{j1}\right| .$$

The Pearson correlation distance between Y_i and Y_j is

$$d_{\mathrm{R}}\left(Y_i, Y_j\right) = 1 - r_{Y_i Y_j} ,$$

where $r_{Y_i Y_j}$ is the Pearson correlation coefficient between Y_i and Y_j.

The Spearman correlation distance between Y_i and Y_j uses the rank-based correlation coefficient in which the expression levels are replaced with the ranks.

More definitions can be found in the book by *Draghici* [33.30]. We should note that the Euclidean and city-block distance look for similar expression numerical values while the Pearson and Spearman distances tend to emphasize similar expression patterns.

The distances defined above measure the gene-wise distance. When clusters are found, we also need to define the distance between two clusters. The four approaches are: single linkage distance (the minimum distance between any gene in one cluster and any gene in the other cluster), complete linkage distance (the maximum distance between any gene in one cluster and any gene in the other cluster), average linkage distance (the average of all pair-wise distances between any gene in one cluster and any gene in the other cluster), and centroid linkage distance (the distance between the centroids of the two clusters).

Clustering Methods

When a distance measure is chosen, there are different ways to execute the clustering process. The clustering methods broadly fall into two categories: hierarchical methods and partitioning methods. Hierarchical methods build up a hierarchy for clusters, from the lowest one (all genes are in one cluster) to the highest one (all genes are in their own clusters) while partitioning methods group the genes into the different clusters based on their expression profiles. Therefore, one does not need to provide the cluster number for hierarchical clustering methods but it is necessary for the partitioning clustering methods.

Hierarchical methods include agglomerative hierarchical methods and divisive hierarchical methods.

The agglomerative hierarchical methods use a bottom-up strategy by treating each individual gene as a cluster at the first step. Then two nearest genes are found and assigned into a cluster where the *nearest* is defined by the distance between these two genes, e.g., for a Pearson distance nearest means the two genes having the largest correlation coefficient. Then an agglomerative hierarchical method assigns a new expression profile for the formed clusters, and repeats these steps until there is only one cluster left.

The divisive hierarchical methods, on the other hand, treat all genes belonging to one cluster at the beginning. Then in each step they choose a partitioning method to divide all genes into a predecided number of clusters, e.g., using k-means to partition genes into two clusters at each single step. Therefore, the decisive hierarchical clustering methods employ the bottom-down strategy.

The k-means clustering is the simplest and fastest clustering algorithm [33.42] among the partitioning methods. It has been widely used in many microarray analyses. To form K clusters, the k-means algorithm allocates the observations into different groups in order to minimize the within-group sum of squares

$$\min_{S_K}\left[\sum_{k=1}^{K}\sum_{i\in S_K}\sum_{j=1}^{m}\left(y_{ij} - \overline{y}_{kj}\right)^2\right] ,$$

where K is the prespecified cluster number, S_k is the set of objects in the k-th cluster and $\overline{y}_{kj}$ is the mean of group j in cluster k. In other words, k-means clustering uses the Euclidean distance.

The k-means clusters are formed through iterations as follows: First, k center genes are randomly selected, and every other gene is assigned to the closest center gene. Then, the center is redefined for each cluster to

minimize the sum of squares toward the center. In fact, the coordinates of a cluster center are the mean expressions of all the genes in that cluster. After the centers are redefined, all genes are regrouped and the iteration process continues until it converges.

After analyzing a yeast cell-cycle expression dataset, *Duan* and *Zhang* [33.43] noted that it could be particularly useful to use a weighted sum of squares for gene clustering to take into account the loss of synchrony of cells. We refer to *Duan* and *Zhang* [33.43] for the details.

Another widely used partitioning clustering algorithm is *self-organizing maps* (SOMs) which were developed by *Kohonen* [33.44]. In essence, SOM clustering is a spatial version of the k-means clustering. For a prespecified grid (i. e., a 6×8 hexagonal grid), SOMs project high-dimensional gene expression data onto a two- or three-dimensional map and place similar genes close to each other. Here, the centroid positions of clusters are related to one another via a spatial topology (e.g., the squared map), and are also iteratively adjusted according to the data.

Both the k-means and SOMs are algorithmic methods and do not have a probabilistic justification. *Probabilistic model-based* clustering (PMC) analysis, on the other hand, assumes that the data is generated by a mixture of underlying probability distributions, and uses the maximum-likelihood method to estimate parameters that define the number of clusters as well as the clusters. Hence, we do not need to specify the number of clusters. Using the probabilistic model, we can even consider covariates while determining the clustering memberships of the genes. However, the model can quickly become complicated as the number of clusters increases. Thus, we must try to use parsimonious models as much as possible. Finally, PMC and k-means are also closely related. In fact, k-means can be interpreted as a parsimonious model of simple independent Gaussians [33.15, 45, 46].

Example 2: Clustering Analysis. In this example, first we perform a hierarchical clustering analysis on the 490 SDE genes from example 1. The clustering analysis is applied in two directions: clustering on samples and clustering on genes. Although we do not present the entire the clustering tree here, two major clusters are formed to distinguish tumor and normal samples. For clustering on the genes, there are roughly five major patterns in terms of the gene expressions. One pattern corresponds to the underexpressed genes and the other four corresponds to the overexpressed genes in the tumor samples versus the normal ones.

For illustration, we selected the first 10 normal arrays and the first 10 cancer arrays, and 20 overexpressed and 20 underexpressed genes randomly from the 490 SDE genes. Figure 33.3 is from the heatmap function in *R*. Though not perfect, two patterns are formed mostly along the line of normal versus tumor tissues. There are roughly five major patterns in terms of expression profiles. Overexpressed and underexpressed genes tend to belong to different clusters. For example, pattern 3 (P3) and pattern 4 (P4) are mainly composed of underexpressed genes while the other three clusters contain mainly overexpressed genes.

Following the hierarchical clustering analysis presented above, we also applied the k-means approach to the 490 SDE genes and set the number of clusters to five. Furthermore, we applied *probabilistic model-based* clustering (PMC) to the same dataset. We examined the BIC (Bayesian information criterion) for different numbers of clusters, and it turned out that the value of

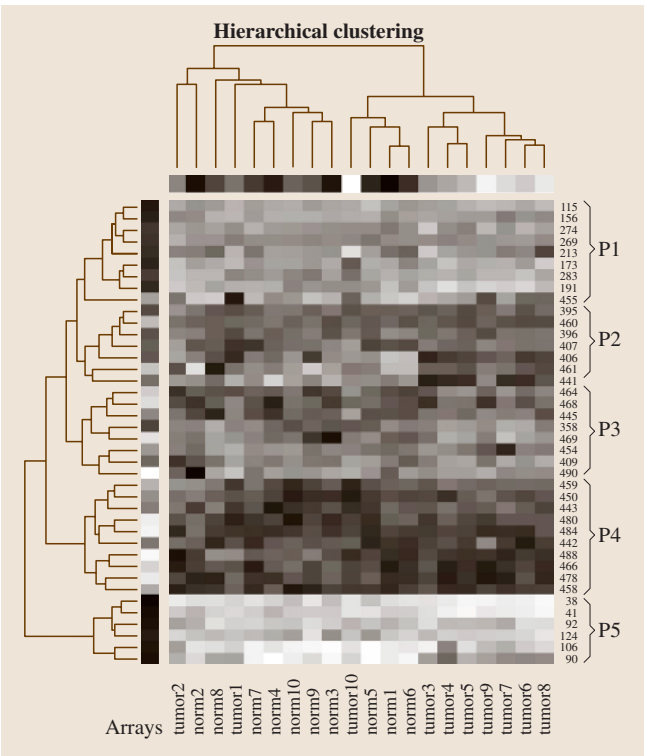


Fig. 33.3 Hierarchical clustering based on a subset of the colon-cancer dataset. Each column corresponds to a sample, and each row a gene. The underexpressed genes were assigned numbers above 440, and the overexpressed genes at or below 440

Table 33.1 The numbers of genes belonging to the intersects of the five k-means clusters and the 13 PMC clusters

k-Means Clusters	PMC Clusters												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	11	0	0	0	57	35	0	0	0	0	29	0
2	25	8	0	0	62	5	0	0	0	0	0	0	0
3	0	0	2	13	0	0	0	0	23	0	0	0	0
4	0	0	15	0	0	0	0	0	1	31	0	0	41
5	0	2	0	0	0	0	1	65	0	34	24	6	0

BIC reaches its minimum at 13 clusters, which is much more than heuristic choice of five. Table 33.1 displays the numbers of genes belonging to the intersects of the five k-means clusters and the 13 PMC clusters. Each of the five k-means clusters is a union of four or so PMC clusters. In fact, if we choose five PMC clusters, they are very similar to the formation of the five k-means clusters, and we will assess this similarity in the next section.

Measure of Agreement Between Two Sets of Clusters

From both the methodological and biologic points of view, there is a need to compare the clusters from different clustering methods. For example, to evaluate the performance of a new clustering approach, we need to compare the derived clusters with the underlying membership in a simulation study. We may also be interested in comparing clustering results derived from the same mRNA samples but being hybridized and analyzed in two different laboratories.

A commonly used measure of agreement between two sets of clusters is the so-called adjusted Rand index (ARI) [33.15, 47, 48]. Let us consider the partitions U and V, and let $\boldsymbol{n}_{ij}$ be the number of genes falling in the intersect of the i-th cluster in U and the j-th cluster in V. The ARI is defined as

$$\frac{\sum_{i,j}\binom{n_{ij}}{2}-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i\cdot}}{2}+\sum_j\binom{n_{\cdot j}}{2}\right]-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}},$$

where $n_{i\cdot}$ and $n_{\cdot j}$ are the numbers of genes in the i-th cluster of U and the j-th cluster of V, respectively.

We suggested some similarity between the k-means and PMC clusters. In fact, the ARI value between the two sets of clusters is 0.425, and it increases to 0.94 if both methods use five clusters. This similarity is expected, because PMC and k-means are equivalent if PMC assumes an independent Gaussian covariance structure [33.15].

33.2.2 Classification

In most microarray experiments, we know the groups on the arrays. For example, some mRNA samples were extracted from tumor cells and the others from normal cells. This is similar to the situation in Sect. 33.1.1. Therefore, it is natural to use this information in analysis and to class cells based on the expression profiles. This is so-called supervised learning.

In Sect. 33.1.1, $Y_{ij,k}$ denotes the expression level of the i-th gene on the j-th array in the k-th sample. Here, we also use $(Y_{ij}, Z = k)$ to reflect the fact that the expression level Y_{ij} of the i-th gene on the j-th array comes from the k-th sample. In other words, the sample group is represented by Z, which is the response or dependent variable in classification.

The essence of classification is to define domains in the feature space spanned by Y_{ij} and to assign a class membership Z to each domain. Classification methods differ in the choice of the shape for the domain and in the algorithm to identify the domain. Some elementary concepts are useful to distinguish these differences. The first one is *linearity*. It refers to a linear combination of the features (expressions of different genes) that forms a hyperplane separating different domains in the feature space. The second term is *separability*. It reflects the extent that the different classes of samples are separable. The third concept is *misclassification*. Often, data are only partially separable, and misclassification is inevitable. In this circumstance, we may need to define a cost function to accommodate different classification errors.

In the machine-learning literature, there is also a distinction between the *learning* (i.e., training) and the *test* samples. The learning data are used to train the classification algorithm and the test data are used to test the predictive ability of the trained classification algorithm. In practice, however, we usually have one dataset and have to split the sample into the training and test samples by leaving a portion of data out during the learning process and saving it as the test data. This pro-

cedure is called cross-validation. More precisely, for a v-fold cross-validation, we first divide the data into v approximately equal sub-samples. Then, we use $v-1$ sub-samples as the training data to construct a classification rule and the left-over subsample as the test data to validate the classification rule. After rotating every sub-sample between training and test data, the performance of the classification rule is assessed through the average in the v runs of validation in the test sample.

In the next subsections, we will review four classification methods that are useful for classifying tissue samples based on gene expression profiles. The methods are linear discriminant analysis (LDA), support vector machines (SVM), artificial neural networks (ANN), and tree-based classification.

LDA

LDA was introduced by Fisher in 1936 for classifying samples by finding a hyperplane that maximizes the between-class variances. Let S_Y be the common sample covariance matrix of all gene expressions, $\bar{Y}_1$ and $\bar{Y}_2$ be the average expression levels of the genes in groups 1 and 2, respectively. The solution to LDA is $S_Y^{-1}(\bar{Y}_1 - \bar{Y}_2)$.

SVM

SVM was first proposed by *Boser* et al. [33.49] and *Cortes* and *Vapnik* [33.50]. SVM finds an optimal hyperplane to separate samples and to allow the maximum *separation* between different classes of samples. The margin of the region that separates samples is *supported* by a few vectors, termed support vectors.

In a two-class classification problem, let $Z = 1$ or -1 denote the two classes. If the two classes of samples are separable, we find a hyperplane $\{y : y^{\mathrm{T}}\beta + \beta_0 = 0, ||\beta|| = 1\}$ such that $(y^{\mathrm{T}}\beta + \beta_0)Z \geq C \geq 0$, where C is the margin optimized to allow the maximal space between the two classes of samples.

For nonseparable case, the procedure is much complicated. Some points will inevitably be on the wrong side of the hyperplane. The idea is to introduce a slack variable to reflect how far a sample is on the wrong side, and then look for the hyperplane at the condition of the total misclassification less than a user-selected limit (i. e., bound the sum of slack variables by a constant). We refer to *Vapnik* [33.51] for the details.

Example 3: Support Vector Machine (SVM). In this example, we perform a classification analysis on the colon-cancer data by SVM. We use M26697 and M63391, the two most significant genes that were identified by SAM from example 1. Specifically, M26697 is the most significant overexpressed gene and M63391 is the most significant underexpressed gene. We used the SVM function in *R* with the cost equal to 100, γ of 1 and tenfold cross-validation, where γ is the coefficient of the radial kernel used to form a hyperplane. Figure 33.4 displays the contour plot of the SVM result. The prediction model correctly classifies 37 cancer and 20 normal samples, but misclassifies three cancer and two normal samples.

Neural Network

The artificial neural network (ANN) is a very popular methodology in machine learning. Also referred to as connectionist architectures, parallel distributed processing, and neuromorphic systems, ANN is an information-processing paradigm with collections of mathematical models that emulate the densely interconnected, parallel structure of the mammalian brain and adaptive biological learning. It is composed of a large number of highly interconnected processing elements that are analogous to neurons and are tied together with

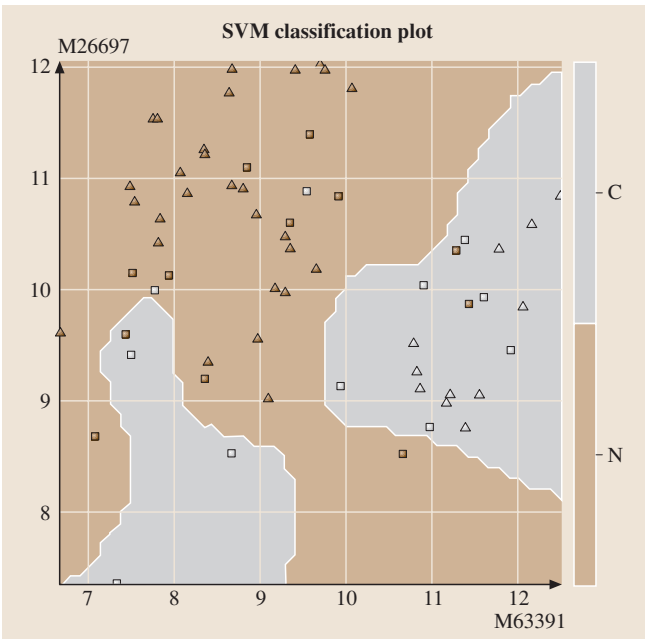


Fig. 33.4 Contour plot of the SVM result using two genes: M26997 and M63391 for the colon-cancer data. C represents cancer and N represents normal. The *light-gray area* is the cancer region and the *brown area* is the normal region. *Square points* represent the support vectors and the *triangle points* represent the data points other than support vectors. The *brown* and *white points* belong to the cancer and the normal regions, respectively

weighted connections that are analogous to synapses. Learning typically occurs by example through training, or exposure to a true set of input/output data where the training algorithm iteratively adjusts the connection weights (synapses). These connection weights store the knowledge necessary to solve specific problems.

ANN can be used for feature selection and feature extraction. The former amounts to variable selection and reduction in statistics and the latter is a generation of the statistical techniques such as principal component analysis, factor analysis, and linear discriminant analysis that are intended to identify lower-dimensional data structures such as linear directions. These lower-dimensional structures usually depend on all of the original variables (i. e., features). Thus, ANN is in essence a computationally intensive version of traditional statistical methods such as regression, classification, clustering, and factor analysis. However, ANN is designed in a way that mimics neural networks and is biologically intuitive and appealing in many applications. This is the major reason that we plan to consider ANN as one of the primary tools to explore the unknown relationship in our data, which is usually referred to as pattern recognition.

The advantage of ANNs lies in their resilience against distortions in the input data and their capability for learning. They are often good at solving problems that are too complex for conventional technologies (e.g., problems that do not have an algorithmic solution, or for which an algorithmic solution is too complex to be found), and are often well-suited to problems that people are good at solving, but for which traditional methods are not.

There are multitudes of different types of ANNs. Some of the more popular include the multilayer perceptron, which is generally trained with the back-propagation of error algorithm, learning vector quantization, radial basis functions, Hopfield, and Kohonen, to name a few. Some ANNs are classified as feed-forward while others are recurrent (i. e., implement feedback) depending on how data is processed through the network. Some ANNs employ supervised training while others are referred to as unsupervised or self-organizing.

Figure 33.5 illustrates a conventional three-layer neural network with n features and K classes. For this feed-forward neural network, the inputs are $y_1, \cdots, y_n$ which correspond to the gene expression profiles and the outputs are $z_1, \cdots, z_K$, which correspond to the K samples in the microarray data. The middle layer consists of many hidden units (also called neurons) and the number of hidden units can be freely chosen and determine the maximum nonlinearity. Each line in Fig. 33.5 indicates a weight—the edge—in the network. This weight represents how much the two neurons which are connected by it can interact. If the weight is larger, then the two neurons can interact more, that is, a stronger signal can pass through the edge. The nature of the interconnections between two neurons can be such that one neuron can either stimulate (a positive weight α) or inhibit (a negative weight α) the other. More precisely, in each hidden unit, we have

$$X_m = \sigma\left(\alpha_{0m} + \alpha_m^{\mathrm{T}} Y\right) ,$$

where σ is called the activation function or neural function and $(\alpha_{0m}, \alpha_m^{\mathrm{T}})$ are the weights. A common choice for σ is the sigmoid function,

$$\sigma(\upsilon) = \frac{1}{1 + \mathrm{e}^{-\upsilon}} .$$

The output function allows a final transformation of the linear combinations of the hidden unit variables,

$$f_k(z) = g_k\left(\beta_{0k} + \beta_k^{\mathrm{T}} X\right) .$$

For a K-class classification, a softmax (logistic) function is usually chosen for the output function

$$g_k(T) = \frac{\mathrm{e}^{T_k}}{\sum_{l=1}^{K} \mathrm{e}^{\mathrm{T}_l}} .$$

During the training period we present the perceptron with inputs one at a time and see what output it gives. If

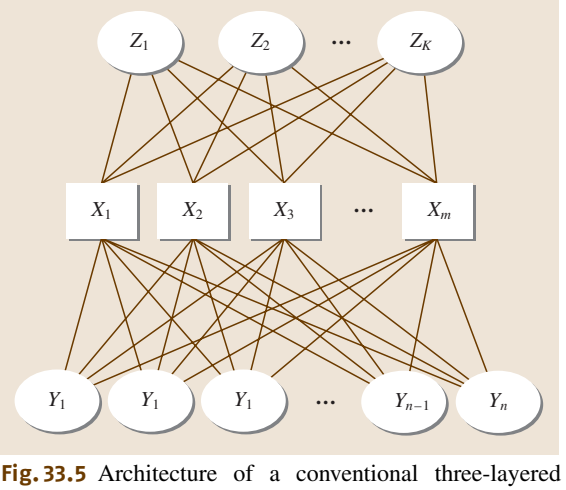


Fig. 33.5 Architecture of a conventional three-layered feed-forward neural network

the output is wrong, we will tell it that it has made a mistake. It should then change its weights and/or threshold properly to avoid making the same mistake later.

33.2.3 Tree- and Forest-Based Classification

One of the most convenient and intuitive approaches for classification is classification trees [33.52, 53]. Classification trees, and their expansion to forests, are based on the so-called recursive partitioning technique. The basic idea of recursive partitioning is to extract homogeneous strata of the tissue samples through expression profiles depending on the expression levels of a particular gene.

Zhang and *Yu* [33.54] reanalyzed the dataset from *Hedenfalk* et al. [33.55] to classify breast cancer mutations in either the BRCA1 or BRCA2 gene using gene expression profiles. *Hedenfalk* et al. [33.55] collected and analyzed biopsy specimens of primary breast cancer tumors from seven and eight patients with germline mutations of BRCA1 and BRCA2, respectively. In addition, seven patients with sporadic cases of primary breast cancer whose family history was unknown were also identified. They obtained cDNA microarrays from 5361 unique genes, of which 2905 are known genes and 2456 are unknown. Thus, in this dataset, Let $Z = 1, 2, 3$ denote BRCA1, BRCA2, and sporadic cases, respectively.

If we use this entire breast cancer dataset to construct a tree, these 22 samples form the initial learning sample, which is called the root node and labeled as node 1 in the tree diagram (Fig. 33.6). The tree structure is determined by recursively selecting a split to divide an upper layer node into two offspring nodes. To do this, we need to evaluate the homogeneity, or the impurity to its opposite,

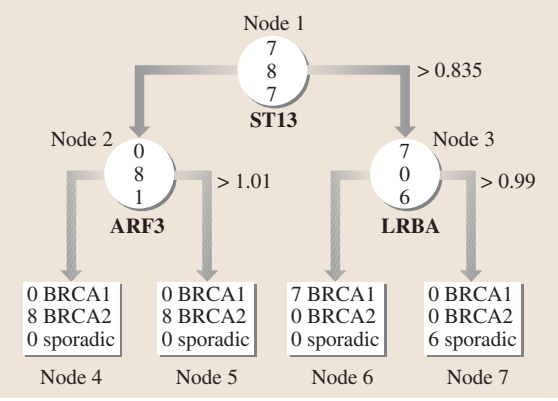


Fig. 33.6 Classification tree for breast-cancer data

of any node. A common measure of node impurity is the entropy function,

$$i_t = -\sum_{k=1}^{K} P(Z = k|\text{node } t) \log[P(Z = k|\text{node } t)] .$$

If node t is the root node, then $P(Z = 1|\text{node } t) = 7/22$, $P(Z = 2|\text{node } t) = 8/22$, and $P(Z = 3|\text{node } t) = 7/22$. Thus, the impurity i_t of the root node can be calculated easily as follows: $i_t = -(7/22)\log(7/22) - (8/22)\log(8/22) - (7/22)\log(7/22) = 1.097$.

How good is the root node? The impurity is zero for a perfect node in which $P(Z = k|\text{Node } t)$ is either 0 or 1, and reaches its worst level when $P(Z = k|\text{node } t) = \frac{1}{3}$ with $i_t = 1.099$. Therefore, the impurity of the root node is near the worst level by design, motivating us to partition the root node into small nodes to reduce the impurity.

The first step of the recursive partitioning process is to divide the root of 32 samples in Fig. 33.6 into two nodes, namely, nodes 2 and 3 in Fig. 33.6. There are many ways of partitioning the root node, because we can take any of the 5361 genes and split the root node according to whether the expression level of this chosen gene is greater than any threshold c. After comparing all possible partitions, we choose the gene and its threshold to keep both i_2 in node 2 and i_3 in node 3 at their lowest possible levels simultaneously. Mathematically, we achieve this goal by minimizing the weighted impurity $r_2 i_2 + r_3 i_3$, where r_2 and r_3 are the proportions of tissue samples in nodes 2 and 3, respectively. This is precisely how the first split (i. e., whether ST13 > 0.835) in Fig. 33.6 is determined.

Once the root is split into nodes 2 and 3, and we can apply the same procedure to potentially split nodes 2 and 3 further. Indeed, the tree in Fig. 33.6 divides the 22 samples into four groups using Heping Zhang's RTREE (http://peace.med.yale.edu). Nodes 2 and 3 are divided based on the expression levels of genes ARF3 and LRBA.

Using a variety of analytic techniques including a modified F- and t-test and a mutual-information scoring, *Hedenfalk* et al. [33.55] selected nine differentially expressed genes to classify BRCA1-mutation-positive and negative tumors and then 11 genes for BRCA2-mutation-positive and negative tumors. Clearly, the tree in Fig. 33.6 uses fewer genes and is a much simpler classification rule.

Although Fig. 33.6 is simple, it does not contain the potentially rich information in the dataset. To improve the reliability of the classification and to accommo-

date potentially multiple biological pathways, *Zhang* and *Yu* [33.54] and *Zhang* et al. [33.56] proposed expanding trees to forests. The large number of genes in microarray makes it an ideal application for these forests.

The most common approach to constructing forests is to perturb the data randomly, form a tree from the perturbed data, and repeat this process to form a series of trees; this is called a random forest. After a forest is formed, we aggregate information from the forest. One such scheme, called bagging (bootstrapping and aggregating), generates a bootstrap sample from the original sample. The final classification is then based on the majority vote of all trees in the forest [33.57].

It is well-known that random forests [33.18, 57] improve predictive power in classification. After observing the fact that there are typically many trees that are of equally high predictive quality in analyzing genomic data, *Zhang* et al. [33.18] proposed a method to construct forests in a deterministic manner. Deterministic forests eliminate the randomness in the random forests and maintain a similar, and sometimes improved, level of precision as the random forests.

The procedure for constructing the deterministic forests is simple. We can search and collect all distinct trees that have a nearly perfect classification or are better than any specified precision. This can be carried out by ranking the trees in deterministic forests. One limitation for the forests (random or deterministic) is that we cannot view all trees in the forests. However, we can examine the frequency of genes as they appear in the forests. Frequent and prominent genes may then be used and analyzed by any method as described above. In other words, forest construction offers a mechanism for data reduction. For the breast-cancer data, one of the most prominent genes identified in the forests is ERBB2. *Kroll* et al. [33.58] analyzed the gene expression patterns of four breast-cancer cell lines: MCF-7, SK-BR-3, T-47D, and BT-474, and reported unique high levels of expressions in the receptor tyrosine kinase ERBB2.

33.3 Fourth-Level Analysis of Microarray Data

Nowadays, different types and platforms of microarray have been developed to address the same (or similar) biological problems. How to integrate and exchange the information contained in different sources of studies effectively is an important and challenging topic for both biologists and statisticians [33.59]. The strategy depends on the situation. When all studies of interest were conducted under the same experimental conditions, this is a standard situation for meta-analysis. There are situations where the experiments are similar, but different platforms were measured, such as the integration of one cDNA array-based study and one oligonucleotide array-based study. There are also situations where different biomolecule microarrays were collected, such as the integration of a genomic array study and a proteomic array study.

Integrating a cDNA array and an Affymetrix chip is complicated because genes on a cDNA array may correspond to several genes (or probesets) on the Affymetrix chip based on the Unigene cluster-matching criteria [33.60]. Instead of matching by genes, matching by the sequence-verified probes may increase the correlation between two studies [33.61].

Most meta-analyses of microarray data have been performed in a study-by-study manner. For example, *Yauk* et al. [33.62] use the Pearson coefficient to measure the correlation across studies, *Rhodes* et al. [33.63] and *Wang* et al. [33.64] use the estimations from one study as prior knowledge while analyzing other studies, and *Welsh* et al. [33.65] treat DNA microarrays as a screening tool and then use protein microarrays to identify the biomarker in cancer research. While they are convenient, these strategies are not ideal [33.63, 66]. Thus, it is imperative and useful to develop better methods to synthesize information from different genomic and proteomic studies [33.59, 62, 67].

33.4 Final Remarks

The technology of gene and protein chips is advancing rapidly, and the entire human genome can be simultaneously monitored on a single chip. The analytic methodology is evolving together with the technology development, but is far from satisfactory. This article reviews some of the commonly used methods in ana-

lyzing microarrays. Analyzing microarray data is still challenging; some of the important issues include how to interpret the results in the biological context, how to improve the reproducibility of the conclusions, and how to integrate information from related but different studies.

References

33.1 M. Schena, M. Shalon, R.W. Davis, P.O. Brown: Quantitative monitoring of gene-expression patterns with a complementary-DNA microarray, Science **270**, 467–470 (1995)

33.2 R.A. Heller, M. Schena, A. Chai, D. Shalon, T. Bedilion, J. Gilmore, D.E. Woolley, R.W. Davis: Discovery and analysis of inflammatory disease-related genes using cDNA microarrays, Proc. Natl. Acad. Sci. USA **94**(6), 2150–2155 (1997)

33.3 E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, N. Friedman: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data, Nature Genetics **34**, 166–176 (2003)

33.4 J.C. Hacia, B. Sun, N. Hunt, K. Edgemon, D. Mosbrook, C. Robbins, S.P.A. Fodor, D.A. Tagle, F.S. Collins: Strategies for mutational analysis of the large multiexon ATM gene using high-density oligonucleotide arrays, Genome Res. **8**, 1245–1258 (1998)

33.5 J.B. Fan, X.Q. Chen, M.K. Halushka, A. Berno, X.H. Huang, T. Ryder, R.J. Lipshutz, D.J. Lockhart, A. Chakravarti: Parallel genotyping of human SNPs using generic high-density oligonucleotide tag arrays, Gen. Res. **10**, 853–860 (2000)

33.6 S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, T.R. Golub: Multiclass cancer diagnosis using tumor gene expression signatures, Proc. Natl. Acad. Sci. USA **98**, 15149–15154 (2001)

33.7 E.R. Marcotte, L.K. Srivastava, R. Quirion: DNA microarrays in neuropsychopharmacology, Trends Pharmacol. Sci. **22**, 426–436 (2001)

33.8 C. Li, W.H. Wong: Model-based analysis of oligonucleotide arrays: expression index computation, outlier detection, Proc. Natl. Acad. Sci. USA **98**, 31–36 (2001)

33.9 B. Efron, R. Tibshirani, J.D. Storey, V. Tusher: J. Amer. Stat. Assoc **96**, 1151–1160 (2001)

33.10 V.G. Tusher, R. Tibshirani, G. Chu: Significance analysis of microarrays applied to the ionizing radiation response, Proc. Natl. Acad. Sci. USA **98**, 5116–5121 (2001)

33.11 R.A. Irizarry, B. Hobbs, F. Collin, Y.D. Beazer-Barclay, K.J. Antonellis, U. Scherf, T.P. Speed: Exploration, normalization, and summaries of high density oligonucleotide array probe level data, Biostat. **4**, 249–264 (2003)

33.12 M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein: Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA **95**, 14863–14868 (1998)

33.13 A. Soukas, P. Cohen, N.D. Socci, J.M. Friedman: Leptin-specific patterns of gene expression in white adipose tissue, Genes Dev. **14**(8), 963–980 (2000)

33.14 P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, T.R. Golub: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc. Natl. Acad. Sci. USA **96**(6), 2907–2912 (1999)

33.15 K.Y. Yeung, W.L. Ruzzo: Principal component analysis for clustering gene expression data, Bioinformatics **17**, 763–774 (2001)

33.16 K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, W.L. Ruzzo: Model-based clustering and data transformations for gene expression data, Bioinformatics **17**, 977–987 (2001)

33.17 O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman: Missing value estimation methods for DNA microarrays, Bioinformatics **17**(6), 520–525 (2001)

33.18 H.P. Zhang, C. Yu, B. Singer: Cell and tumor classification using gene expression data: construction of forests, Proc. Natl. Acad. Sci. USA **100**, 4168–4172 (2003)

33.19 T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler: Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics **16**(10), 906–914 (2000)

33.20 K. Mehrotra, C.K. Mohan, S. Ranka: *Elements of Artificial Neural Networks* (MIT, Massachusetts 1997)

33.21 H.P. Zhang, C. Yu, B. Singer, M. Xiong: Recursive partitioning for tumor classification with gene expression microarray data, Proc. Natl. Acad. Sci. USA **98**, 6730–6735 (2001)

33.22 A.J. Butte, P. Tamayo, D. Slonim, T.R. Golub, I.S. Kohane: Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks, Proc. Natl. Acad. Sci. USA **97**, 12182–12186 (2000)

33.23 P. D'haeseleer, S. Liang, R. Somogyi: *Gene expression data analysis and modeling* (Pacific Symposium on Biocomputing, 1999)

33.24 I. Shmulevich, E.R. Dougherty, S. Kim, W. Zhang: Probabilistic Boolean networks: a rule-based un-

certainty model for gene regulatory networks, Bioinformatics **18**(2), 261–274 (2002)
33.25 N. Friedman, M. Linial, I. Nachman, D. Pe'er: Using Bayesian networks to analyze expression data, J. Comp. Biol. **7**, 601–620 (2000)
33.26 E. Segal, B. Taskar, A. Gasch, N. Friedman, D. Koller: Rich probabilistic models for gene expression, Bioinformatics **1**, 1–10 (2001)
33.27 D.J. Lockhart, H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, E.L. Brown: Expression monitoring by hybridization to high-density oligonucleotide arrays, Nat. Biotechnol. **14**, 1675–1680 (1996)
33.28 G. Smyth: Linear models and empirical Bayes methods for assessing differential expression in microarray experiments, Stat. Appl. Genet. Mol. Biol, **3**(1), 3 (2004)
33.29 Z. Šidák: Rectangular confidence regions for the means of multivariate normal distributions, J. Am. Stat. Assoc. **62**, 626–633 (1967)
33.30 S. Draghici: *Data analysis tools for DNA microarrays* (Chapman, Hall/CRC, New York 2003)
33.31 Y. Benjamin, Y. Hochberg: Controlling the false discovery rate – a practical and powerful approach to multiple testing, J. Roy. Soc. B Met. **57**(1), 289–300 (1995)
33.32 J.D. Storey: A direct approach to false discovery rates, J. R. Stat. Ser. B Stat. Methodol. **64**, 479–498 Part 3 (2002)
33.33 J.D. Storey: A Bayesian interpretation, the q-value, Ann. Stat, **31**(6), 2013–2035 (2003)
33.34 J.F. Troendle: Stepwise normal theory multiple test procedures controlling the false discovery rate, J. Stat. Plan. Inference **84**(1-2), 139–158 (2000)
33.35 B. Efron, R. Tibshirani: Empirical bayes methods and false discovery rates for microarrays, Genet. Epidemiol. **23**(1), 70–86 (2002)
33.36 I. Lonnstedt, T. Speed: Replicated microarray data, Stat. Sinica **12**(1), 31–46 (2001)
33.37 U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine: Broad patterns of gene expression revealed by clustering analysis of tumor, normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA **96**, 6745–6750 (1999)
33.38 J. Quackenbush: Computational analysis of microarray analysis, Nature Rev. Genetics **2**, 418–427 (2001)
33.39 N. Kaminski, N. Friedman: Practical approaches to analyzing results of microarray experiments, Am. J. Respir. Cell. Mol. Biol. **27**(2), 125–132 (2002)
33.40 R. Jansen, D. Greenbaum, M. Gerstein: Relating whole-genome expression data with protein-protein interactions, Genome Res. **12**(1), 37–46 (2002)
33.41 J.C. Boldrick, A.A. Alizadeh, M. Diehn, S. Dudoit, C.L. Liu, C.E. Belcher, D. Botstein, L.M. Staudt, P.O. Brown, D.A. Relman: Stereotyped and specific gene expression programs in human innate immune responses to bacteria, Proc. Natl. Acad. Sci. USA **99**, 972–977 (2002)
33.42 G. Sherlock: Analysis of large-scale gene expression data, Curr. Opin. Immunol. **12**(2), 201–205 (2000)
33.43 F.H. Duan, H.P. Zhang: Correcting the loss of cell-cycle synchrony in clustering analysis of microarray data using weights, Bioinformatics **20**(11), 1766–1771 (2004)
33.44 T. Kohonen: *Self-Organizing Maps* (Springer, Brelin Heidelberg New York 1997)
33.45 W.N. Venables, B.D. Ripley: *Modern Applied Statistics with S* (Springer, Berlin Heidelberg New York 2002)
33.46 E. Wit, J. McClure: *Statistics for Microarrays* (Wiley, New York 2004)
33.47 L. Hubert, P. Arabie: Comparing partitions, J. Classification **2**, 193–218 (1985)
33.48 G.W. Milligan, M.C. Cooper: A study of the comparability of external criteria for hierarchical cluster-analysis, Multivairate Behavioral Research **21**(4), 441–458 (1986)
33.49 B.E. Boser, I.M. Guyon, V.N. Vapnik: A training algorithm for optimal margin classifiers. In: *Fifth Annual Workshop on Computational Learning Theory*, ed. by D. Haussle (ACM, New York 1992) pp. 144–152
33.50 C. Cortes, V. Vapnik: Support-vector networks, Mach. Learn. **20**(3), 273–297 (1995)
33.51 V. Vapnik: *Statistical Learning Theory* (Wiley, New York 1998)
33.52 L. Breiman, J. Friedman, C. Stone, R. Olshen: *Classification, Regression Trees* (Wadsworth, Belmont 1984)
33.53 H.P. Zhang, B. Singer: *Recursive Partitioning in the Health Sciences* (Springer, Berlin Heidelberg New York 1999)
33.54 H. Zhang, C.-Y. Yu: Tree-based analysis of microarray data for classifying breast cancer, Front. in Biosci. **7**, c63–67 (2002)
33.55 I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pittaluga, S. Gruvberger, N. Loman, O. Johannsson, H. Olsson, B. Wilfond, G. Sauter, O.P. Kallioniemi, A. Borg, J. Trent: Gene-expression profiles in hereditary breast cancer, N. Engl. J. Med **344**, 539–48 (2001)
33.56 H.P. Zhang, C.Y. Yu, H.T. Zhu, J. Shi: Identification of linear directions in multivariate adaptive spline models, J. Am. Stat. Assoc. **98**, 369–376 (2003)
33.57 B.L. Random: Random forests, Mach. Learn. **45**, 5–32 (2001)

33.58 T. Kroll, L. Odyvanova, H. Clement, C. Platzer, A. Naumann, N. Marr, K. Hoffken, S. Wolfl: Molecular characterization of breast cancer cell lines by expression profiling, J. Cancer Res. Clin. Oncol. **128**, 125–34 (2002)

33.59 Y. Moreau, S. Aerts, B. D. Moor, B. D. Strooper, M. Dabrowski: Comparison and meta-analysis of microarray data: from the bench to the computer desk, Trends Genetics **9**(10), 570–577 (2003)

33.60 D. Ghosh, T. Barette, D. Rhodes, A. Chinnaiyan: Statistical issues and methods for meta-analysis of microarray data: a case study in prostate cancer, Funct. Integrat. Gen. **3**(4), 180–188 (2003)

33.61 B. H. Mecham, G. T. Klus, J. Strover, M. Augustus, D. Byrne, P. Bozso, D. Z. Wetmore, T. J. Mariani, I. S. Kohane, Z. Szallasi: Sequence-matched robes produce increased cross-platform consistency and more reproducible biological results in microarray-based gene expression measurements, Nucleotide Acids Res. **32**(9), e74 (2004)

33.62 C. L. Yauk, M. L. Berndt, A. Williams, G. R. Douglas: Comprehensive comparison of six microarray technologies, Nucleic Acids Res. **32**(15), e124 (2004)

33.63 D. R. Rhodes, T. R. Barrette, M. A. Rubin, D. Ghosh, A. M. Chinnaiyan: Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer, Cancer Res. **62**(15), 4427–4433 (2002)

33.64 J. Wang, K. R. Coombes, W. E. Highsmith, M. J. Keating, L. V. Abruzzo: Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells, Bioinformatics **20**(17), 3166–3178 (2004)

33.65 J. B. Welsh, L. M. Sapinoso, S. G. Kern, D. A. Brown, T. Liu, A. R. Bauskin, R. L. Ward, N. J. Hawkins, D. I. Quinn, P. J. Russell, R. L. Sutherland, S. N. Breit, C. A. Moskaluk, H. F. Frierson Jr., G. M. Hampton: Large-scale delineation of secreted protein biomarkers overexpressed in cancer tissue and serum, Proc. Natl. Acad. Sci **100**(6), 3410–3415 (2003)

33.66 L. V. Hedges, I. Olkin: *Statistical Methods For Meta-Analysis* (Academic, New York 1985)

33.67 A. K. Järvinena, S. Hautaniemib, H. Edgrena, P. Auvinend, J. Saarelaa, O. P. Kallioniemic, O. Monni: Are data from different gene expression microarray platforms comparable?, Genomics **83**(6), 1164–1168 (2004)

34. Statistical Methods in Proteomics

Proteomics technologies are rapidly evolving and attracting great attention in the post-genome era. In this chapter, we review two key applications of proteomics techniques: disease biomarker discovery and protein/peptide identification. For each of the applications, we state the major issues related to statistical modeling and analysis, review related work, discuss their strengths and weaknesses, and point out unsolved problems for future research.

We organize this chapter as follows. Section 34.1 briefly introduces mass spectrometry (MS) and tandem MS/MS with a few sample plots showing the data format. Section 34.2 focuses on MS data preprocessing. We first review approaches in peak identification and then address the problem of peak alignment. After that, we point out unsolved problems and propose a few possible solutions.

Section 34.3 addresses the issue of feature selection. We start with a simple example showing the effect of a large number of features. Then we address the interaction of different features and discuss methods of reducing the influence of noise. We finish this section with some discussion on the application of machine learning methods in feature selection. Section 34.4 addresses the problem of sample classification. We describe the random forest method in detail in Sect. 34.5.

In Sect. 34.6 we address protein/peptide identification. We first review database searching methods in Sect. 34.6.1 and then focus on de novo MS/MS sequencing in Sect. 34.6.2. After reviewing major protein/peptide identification programs like SEQUEST and MASCOT in Sect. 34.6.3, we conclude the section by pointing out some major issues that need to be addressed in protein/peptide identification.

Proteomics technologies are considered the major player in the analysis and understanding of protein function and biological pathways. The development of statistical methods and software for proteomics data analysis will continue to be the focus of proteomics for years to come.

34.1 Overview

In the post-genome era, proteomics has attracted more and more attention due to its ability to probe biological functions and structures at the protein level. Although recent years have witnessed great advancement in the collection and analysis of gene expression microarray data, proteins are in fact the functional units that

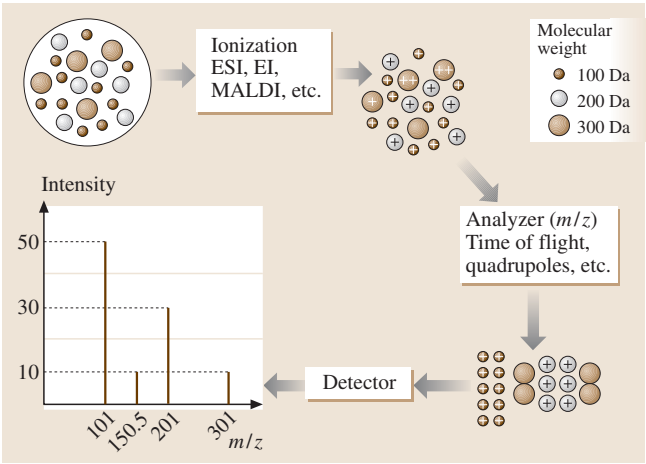


Fig. 34.1 The principle of MS data generation. Molecules are ionized into peptides in the ionizer. The peptides are accelerated and separated by the analyzer and then detected by the detector

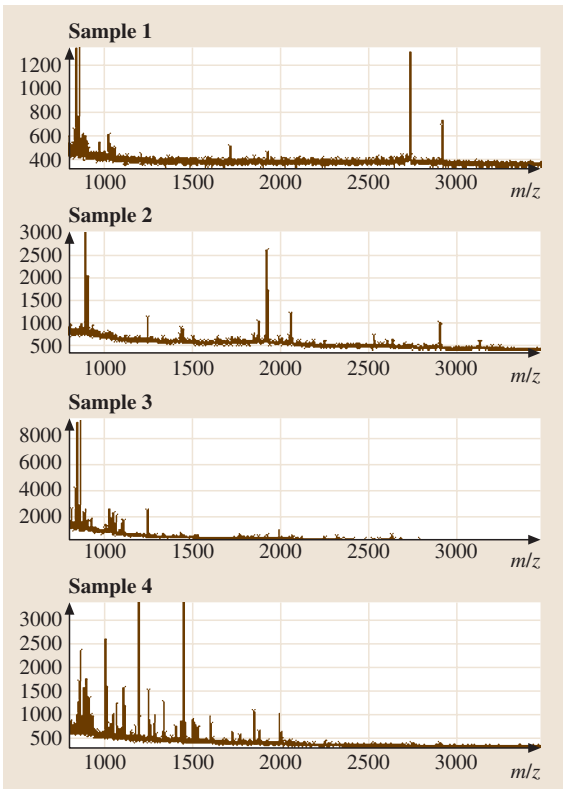


Fig. 34.2 A few examples of MS raw data. The horizontal axis denotes the m/z ratio and the vertical axis denotes the intensity value

are of biological relevance. The often poor correlation that exists between levels of mRNA versus protein expression [34.1], and the rapid advances in mass spectrometry (MS) instrumentation and attendant protein profiling methodologies have substantially increased interest in using MS approaches to identify peptide and protein biomarkers of disease. This great level of interest arises from the high potential of biomarkers to provide earlier diagnosis, more accurate prognosis and disease classification; to guide treatment; and to increase our understanding at the molecular level of a wide range of human diseases. This chapter focuses on two key applications of proteomics technologies: disease biomarker discovery and protein identification through MS data. We anticipate that statistical methods and computer programs will contribute greatly to the discovery of disease biomarkers as well as the identification of proteins and their modification sites. These methods should help biomedical researchers to better realize the potential contribution of rapidly evolving and ever more sophisticated MS technologies and platforms.

The study of large-scale biological systems has become possible thanks to emerging high-throughput mass spectrometers. Basically, a mass spectrometer consists of three components: ion source, mass analyzer, and detector. The ion source ionizes molecules of interest into charged peptides, the mass analyzer accelerates these peptides with an external magnetic field and/or electric field, and the detector generates a measurable signal when it detects the incident ions. This procedure of producing MS data is illustrated in Fig. 34.1. Data resulting from MS sources have a very simple format consisting entirely of paired mass-to-charge ratio (m/z value) versus intensity data points. Figure 34.2 shows a few examples of the raw MALDI-MS data.

The total number of measured data points is extremely large (about 10^5 for a conventional MALDI-TOF instrument, as compared to perhaps 10^6 for a MALDI-FTICR instrument covering the range from 700–28 000 Da), while the sample size is usually on the order of hundreds. This very high ratio of data size to sample size poses unique statistical challenges in MS data analysis. It is desirable to find a limited number of potential peptide/protein biomarkers from the vast amount of data in order to distinguish cases from controls and enable classification of unknown samples. This process is often referred to as biomarker discovery. In this chapter, we review key steps in biomarker discovery: preprocessing, feature selection, and sample classification.

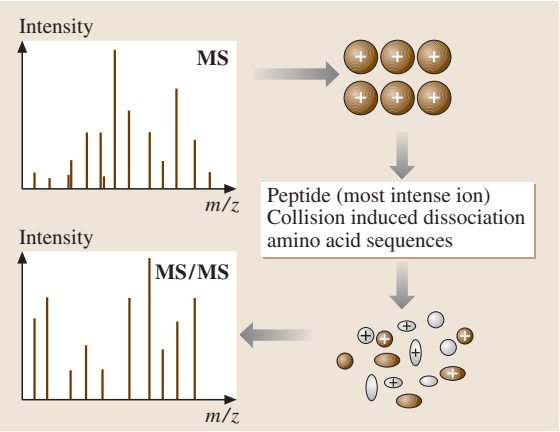


Fig. 34.3 The principle of MS/MS data generation. Peptides are further fragmented through collision-induced dissociation (CID) and detected in the tandem MS equipment

A biomarker discovered in the MS data may correspond to many possible biological sources (so a spectral peak can arise from different proteins). Therefore, it is necessary to identify peptides and their parent proteins in order to fully understand the relation between protein structure and disease development. This understanding can also be very useful in drug design and development.

In order to identify proteins in complex mixtures, the tandem MS technique (MS/MS) coupled with database searching has become the method of choice for the rapid and high-throughput identification, characterization, and quantification of proteins. In general, a protein mixture of interest is enzymatically digested, and the resulting peptides are then further fragmented through collision-induced dissociation (CID). The resulting tandem MS spectrum contains information about the constituent amino acids of the peptides and therefore information about their parent proteins. This process is illustrated in Fig. 34.3.

Many MS/MS-based methods have been developed to identify proteins. The identification of peptides containing mutations and/or modifications, however, is still a challenging problem. Statistical methods need to be developed to improve identification of modified proteins in samples consisting of only a single protein and also in samples consisting of complex protein mixtures.

We organize the rest of the chapter as follows: Section 34.2 describes MS data preprocessing methods. Section 34.3 focuses on feature selection. Section 34.4 reviews general sample classification methodology and Sect. 34.5 mainly describes the random forest algorithm. Section 34.6 surveys different algorithms/methods for protein/peptide identification, each with its strengths and weaknesses. It also points out challenges in the future research and possible statistical approaches to solving these challenges. Section 34.7 summarizes the chapter.

34.2 MS Data Preprocessing

When analyzing MS data, only the spectral peaks that result from the ionization of biomolecules such as peptides and proteins are biologically meaningful and of use in applications. Different data preprocessing methods have been proposed to detect and locate spectral peaks. A commonly used protocol for MS data preprocessing consists of the following steps: spectrum calibration, baseline correction, smoothing, peak identification, intensity normalization and peak alignment [34.2–4].

Preprocessing starts with aligning individual spectra. Even with the use of internal calibration, the maximum observed intensity for an internal calibrant may not occur at exactly the same m/z value in all spectra. This challenge can be addressed by aligning spectra based on the maximum observed intensity of the internal calibrant. For the sample collected, the distance between each pair of consecutive m/z ratios is not constant. Instead, the increment in m/z values is approximately a linear function of the m/z values. Therefore, a log-transformation of m/z values is needed before any analysis is performed so that the scale on the predictor is roughly comparable across the range of all m/z values. In addition to transforming the m/z values, we also need to log-transform intensities to reduce the dynamic range of the intensity values. In summary, log-transformations are needed for both m/z values and intensities as the first step in MS data analysis. Figure 34.4 shows an example of MS data before and after the log-transformation.

Chemical and electronic noise produce background fluctuations, and it is important to remove these background fluctuations before further analysis. Local smoothing methods have been utilized for baseline

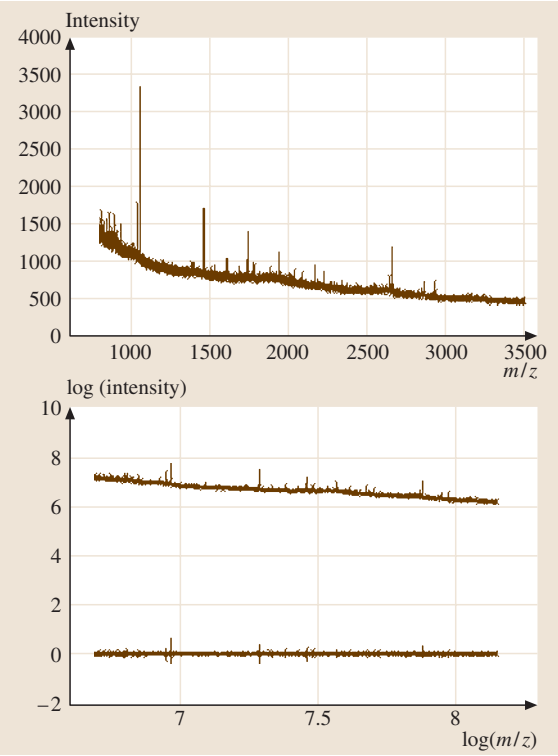


Fig. 34.4 ***Top:*** Original raw data. ***Bottom:*** MS data after the log-transformation (*top*). The result of baseline correction is also shown (*bottom*)

subtraction to remove high frequency noise, which is apparent in MALDI-MS spectra. In the analysis of MALDI data, *Wu* et al. [34.5] used a local linear regression method to estimate the background intensity values, and then subtracted the fitted values from the local linear regression result. *Baggerly* et al. [34.4] proposed a semi-monotonic baseline correction method in their analysis of SELDI data. *Liu* et al. [34.6] computed the convex hull of the spectrum, and subtracted the convex hull from the original spectrum to get the baseline-corrected spectrum. An example of baseline correction is shown in Fig. 34.4 as well.

Among the above steps, peak identification and alignment are arguably the most important ones. The inclusion of non-peaks in the analysis will undoubtedly reduce our ability to identify true biomarkers, while the peaks identified need to be aligned so that the same peptide corresponds to the same peak value.

In the following, we give an overview of the existing approaches related to peak detection and peak alignment.

34.2.1 Peak Detection/Finding

Normally, spectral peaks are local maxima in MS data. Most published algorithms on peak identification use local intensity values to define peaks; in other words peaks are mostly defined with respect to nearby points. For example, *Yasui* et al. [34.3, 7] defined a peak as the m/z value that has the highest intensity value within its neighborhood, where the neighbors are the points within a certain range from the point of interest. In addition, a peak must have an intensity value that is higher than the average intensity level of its broad neighborhood. *Coombes* et al. [34.4] considered two peak identification procedures. For simple peak finding, local maxima are first identified. Then, those local maxima that are likely noise are filtered out, and nearby maxima that likely represent the same peptides are merged. There is a further step needed to remove unlikely peak points. In simultaneous peak detection and baseline correction, peak detection is first used to obtain a preliminary list of peaks, and then the baseline is calculated by excluding candidate peaks. The two steps are iterated and some peaks are further filtered out if the signal-to-noise ratio is smaller than a threshold. Similarly, *Liu* et al. [34.6] declared a point in the spectrum to be a peak if the intensity is a local maximum, its absolute value is larger than a particular threshold, and the intensity is larger than a threshold times the average intensity in the window surrounding this point.

All of these methods are based on similar intuitions and heuristics. Several parameters need to be specified beforehand in these algorithms, such as the number of neighboring points and the intensity threshold value. In fact, the parameter settings in the above algorithms are related to our understanding/modeling of the underlying noise. To address this issue, *Coombes* et al. [34.4] defined noise as the median absolute value of intensity. *Satten* et al. [34.8] used the negative part of the normalized MS data to estimate the variance of the noise. Wavelet-based approaches [34.9, 10] have also been proposed to remove noise in the MS data before peak detection. Based on the observation that there are substantial measurement errors in the measured intensities, *Yasui* and colleagues [34.3] argued that binary peak/non-peak data is more useful than the absolute values of intensity, while they still used a local maximum search method to detect peaks. Clearly, the success of noise-estimation-plus-threshold methodology depends largely on the validity of the noise model, which remains to be seen.

Another issue in peak detection is to avoid false positive detections. This is often done by adding an additional constraint (such as the peak width constraint [34.3]) or by choosing a specific scale level after wavelet decomposition of the original MS data (*Randolph* and *Yasui*) [34.10]. In the case of high-resolution data, it has been proposed that more than one isotopic variant of a peptide peak should be present before a spectral peak is considered to result from peptide ionization (*Yu* et al.) [34.11]. It may also be possible to use prior information about the approximate expected peak intensity distribution of different isotopes arising from the same peptide during peak detection; the theoretical relative abundance of the first peptide isotope peak may range from 60.1% for polyGly (n=23, MW 1329.5 Da) to 90.2% for poly Trp (n=7, MW 1320.5 Da) (personal communication 11/1/04 from Dr. Walter McMurray, Keck Laboratory). Certainly we also have to consider the issue of limited resolution and the consequent overlapping effect of neighboring peaks.

34.2.2 Peak Alignment

After peaks have been detected, we have to align them together before comparing peaks in different data sets. Previous studies have shown that the variation of peak locations in different data sets is nonlinear [34.12, 13]. The example in *Yu* et al. [34.11] shows that this variation still exists even when we use technical replicates. The reasons that underlie data variation are extremely complicated, including differences in sample preparation, chemical noise, cocrystallization and deposition of the matrix-sample onto the MALDI-MS target, laser position on the target, and other factors. Although it is of interest to identify these reasons, we are more interested in finding a framework to reduce the variation and align these peaks together.

Towards this direction, some methods have been proposed. *Coombes* et al. [34.4] pooled the list of detected peaks that differed in location by three clock ticks or by 0.05% of the mass. *Yasui* et al. [34.3] believed that the m/z axis shift of the peaks is approximately $\pm$ 0.1% to $\pm$0.2% of the m/z value. Thus, they expanded each peak to its local neighborhood with the width equal to 0.4% of the m/z value of the middle point. This method certainly oversimplifies the problem. In another study (*Yasui* et al.) [34.7], they first calculated the number of peaks in all samples allowing certain shifts, and selected m/z values using the largest number of peaks. This set of peaks is then removed from all spectra and the procedure is iterated until all peaks are exhausted from all the samples. In a similar spirit, *Tibshirani* et al. [34.14] proposed to use complete linkage hierarchical clustering in one dimension to cluster peaks, and the resulting dendrogram is cut off at a given height. All of the peaks in the same cluster are considered to be the same peak in further analysis.

Randolph and *Yasui* [34.10] used wavelets to represent the MS data in a multiscale framework. They used a coarse-to-fine method to first align peaks at a dominant scale and then refine the alignment of other peaks at a finer scale. From a signal representation point of view, this approach is very interesting. But it remains to be determined whether the multiscale representation is biologically reasonable.

Johnson et al. [34.15] assumed that the peak variation is less than the typical distance between peaks and they used a closest point matching method for peak alignment. The same idea was also used in *Yu* et al. [34.11] to address the alignment of multiple peak sets. Certainly, this method is limited by the data quality and it cannot handle large peak variation.

Dynamic programming (DP) based approaches [34.12, 16] have also been proposed. DP has been used in gene expression analysis to warp one gene expression time series to another similar series obtained from a different biological replicate [34.17], where the correspondence between the two gene expression time series is guaranteed. In MS data analysis, however, the situation is more complicated since a one-to-one correspondence between two data sets does not always exist. Although it is still possible to apply DP to deal with the lack-of-correspondence problem, some modifications are necessary (such as adding an additional distance penalty term to the estimation of correspondence matrix). It also remains unclear how DP can identify and ignore outliers during the matching.

Eilers [34.13] proposed a parametric warping model with polynomial functions or spline functions to align chromatograms. In order to fix warping parameters, he added calibration example sequences into chromatograms. While the idea of using a parametric model is interesting, it is difficult to repeat the same parameter estimation method in MS data since we cannot add many calibrator compounds into the MS samples. Also, it is unclear whether a second-order polynomial would be enough to describe the nonlinear shift in the MS peaks.

Although all of these methods are ad hoc, the relatively small number of peaks (compared to the number of collected points) and the relatively small shifts from spectrum to spectrum ensure that these heuristic peak alignments should work reasonably well in practice.

34.2.3 Remaining Problems and Proposed Solutions

Current peak detection methods (such as the local maximum search plus threshold method) export detection results simply as peaks or non-peaks. Given the noisy nature of MS data, this simplification is prone to being influenced by noise (noise may also produce some local maximal values) and is very sensitive to specific parameter settings (including the intensity threshold value). In addition, a uniform threshold value may exclude some weak peaks in the MS data, while the existence/nonexistence of some weak peaks may be the most informative biomarkers.

Instead of using a binary output, it would be better to use both peak width and intensity information as quantitative measures of how likely it is that a candidate is a true peak. We can use a distribution model to describe the typical peak width and intensity. The parameters of the distribution can be estimated using training samples. Then a likelihood ratio test can be used to replace the binary peak detection result (either as peak (one) or as non-peak (zero)) with a real value. This new measure should provide richer information about peaks. We believe this will help us to better align multiple peak sets.

The challenge in peak alignment is that current methods may not work if we have large peak variation [like with LC/MS (liquid chromatography/mass spectrometry) data]. Another unsolved problem is that it may not be valid to assume that the distribution of peaks is not corrupted by noise (false positive detection).

To address these problems, we may consider the "true" locations of peaks as random variables and regard the peak detection results as sampling observations. Then, the problem of aligning multiple peak sets is converted to the problem of finding the mean (or median) values of random variables since we can assume that the majority of peaks should be located close to the "true" locations, with only a few outliers not obeying this assumption. After the mean/median values have been found/estimated, the remaining task is to simply align peaks w.r.t. the mean/median standard. Intuitively, the relative distance between a peak and its mean/median standard may also be used as a confidence measure in alignment.

34.3 Feature Selection

For current large-scale genomic and proteomic datasets, there are usually hundreds of thousands of features (also called variables in the following discussion) but limited sample size, which poses a unique challenge for statistical analysis. Feature selection serves two purposes in this context: biological interpretation and to reduce the impact of noise.

Suppose we have n_1 samples from one group (e.g. cancer patients) and n_0 samples from another group (e.g. normal subjects). We have m variables $(X_1, \ldots, X_m)$ (e.g. m/z ratios). For the kth variable, the observations are

$$X_k^1 = (x_{k1}, \ldots, x_{kn_1})$$

for the first group and

$$X_k^0 = \left(x_{k(n_1+1)}, \ldots, x_{k(n_1+n_0)}\right)$$

for the second group. They can be summarized in a data matrix, $X = (x_{ij})$. Assume X_k^1 are n_1 i.i.d. samples from one distribution $f_{k1}(x)$ and X_k^0 are n_0 i.i.d. samples from another distribution $f_{k0}(x)$.

Two sample t-test statistics or variants thereof are often used to quantify the difference between two groups in the analysis of gene expression data [34.18–20]

$$T_i = \frac{\bar{x}_{i1} - \bar{x}_{i0}}{\sqrt{\frac{1}{n_1}\hat{\sigma}_{i1}^2 + \frac{1}{n_0}\hat{\sigma}_{i0}^2}}, \tag{34.1}$$

where

$$\bar{x}_{i1} = \sum_{k=1}^{n_1} x_{ik}, \quad \bar{x}_{i0} = \sum_{k=n_1+1}^{n_1+n_0} x_{ik},$$

$$\hat{\sigma}_{i1}^2 = \frac{1}{n_1 - 1}\sum_{k=1}^{n_1}(x_{ik} - \bar{x}_{i1})^2,$$

$$\hat{\sigma}_{i0}^2 = \frac{1}{n_0 - 1}\sum_{k=n_1+1}^{n_1+n_0}(x_{ik} - \bar{x}_{i0})^2.$$

T_i can be interpreted as the standardized difference between these two groups. It is expected that the larger the standardized difference, the more separated the two groups are. One potential problem with using t-statistics is its lack of robustness, which may be a serious drawback when hundreds of thousands of features are being screened to identify informative ones.

34.3.1 A Simple Example of the Effect of Large Numbers of Features

Although there are hundreds of thousands of peaks representing peptides, we expect the number of peaks that provide information on disease classification to be limited. This, coupled with the limited number of samples available for analysis, poses great statistical challenges for the identification of informative peaks. Consider the following simple example: suppose that there are $n_1 = 10, n_2 = 10$, $m_0 = 10^3$ peptides showing no difference, and $m_1 = 40$ peptides showing differences between the two groups with $\lambda = \mu/\sigma = 1.0$. We can numerically calculate the expected number of significant features for these two groups

$$N_0 = 2m_0[1 - T(x, df = 18)] ,$$
$$N_1 = m_1[T(-x, df = 18, \lambda = 1.0) + 1 - T(x, df = 18, \lambda = 1.0)] ,$$

where $T(x, df)$ is the t-distribution function with df degrees of freedom, $T(x, df, \lambda)$ is the t-distribution function with df degrees of freedom and noncentral parameter λ, and the significance cut-off values are chosen as $|T| > x$. Figure 34.5 gives a comparison of true and false positives for this example, where a diagonal line is also shown. We can clearly see the dominant effect of noise in this example.

This artificial example reveals the difficulty that extracting useful features among a large number of noisy features entails. In practice, due to the noisy nature of MS data, the variance σ for individual peptide intensity will be very large, reflecting the difficulties with reproducibility that are commonly observed for MS data. Also, the number of noisy features m_0 (mostly uninformative) are increasing exponentially with the advance of technology (e.g. MALDI-FTICR data). The combination of these two factors will increase the ratio of false/true positives.

In this simple example, we ignore the interaction of different proteins. For complex diseases, such as cancer, it is quite possible that the effects result from the joint synergy of multiple proteins, while they individually show nonsignificant differences. Novel statistical methods are needed to account for the effects of noise and interactions among features.

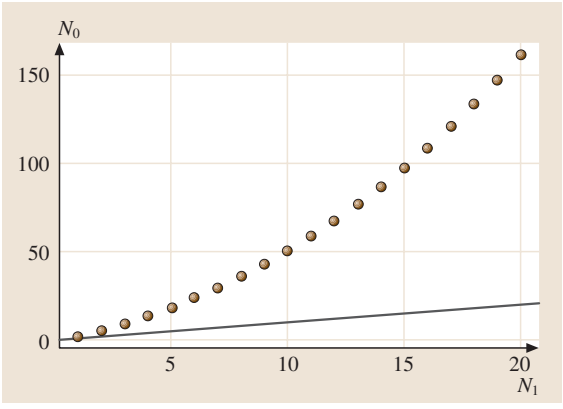


Fig. 34.5 Comparison of true positive and false positive for the simulation example. N_1 is the number of true positive, N_0 false positive. The diagonal line is also plotted as a *solid line*. Different points correspond to different settings of critical values in the t-test

34.3.2 Interaction

In ordinary or logistic regression models, we describe the interaction of different variables by including the interaction terms. This approach quickly becomes unfeasible with an increasing number of variables. Therefore, standard regression models are not appropriate due to $n \ll p$.

Instead of using univariate feature selection methods, it may be useful to consider multivariate feature selection methods. *Lai* et al. [34.21] analyzed the co-expression pattern of different genes using a prostate cancer microarray dataset, where the goal is to select genes that have differential gene–gene coexpression patterns with a target gene. Some interesting genes have been found to be significant and reported to be associated with prostate cancer, yet none of them showed marginal significant differential gene expressions.

Generally, multivariate feature selection is a combinatorial approach. To analyze two genes at a time we need to consider n^2 possibilities instead of n for the univariate feature selection. To analyze the interaction of K genes we need to consider n^K possibilities, which quickly becomes intractable.

A classification and regression tree (CART) [34.22] naturally models the interaction among different variables, and it has been successfully applied to genomic and proteomic datasets where $n \ll p$ is expected [34.23].

There are several new developments that are generalizations of the tree model. Bagging stands for bootstrap aggregating. Intuitively, bagging uses bootstrap to produce pseudoreplicates to improve prediction

accuracy [34.24]. The boosting method [34.25] is a sequential voting method, which adaptively resamples the original data so that the weights are increased for those most frequently misclassified samples. A boosting model using a tree as the underlying classifier has been successfully applied to genomic and proteomic datasets [34.26, 27].

34.3.3 Reducing the Influence of Noise

For most statistical models, the large number of variables may cause an overfitting problem. Just by chance, we may find some combinations of noise that can potentially discriminate samples with different disease status. We can incorporate some additional information into our analysis. For MS data, for instance, we only want to focus on peaks resulting from peptide/protein ionization. In previous sections we have addressed and emphasized the importance of MS data preprocessing.

34.3.4 Feature Selection with Machine Learning Methods

Isabelle et al. [34.28] have reported using SVM to select genes for cancer classification from microarray data. *Qu* et al. [34.29] applied a boosting tree algorithm to classify prostate cancer samples and to select important peptides using MS analysis of sera. *Wu* et al. [34.5] reported using random forest to select important biomarkers from ovarian cancer data based on MALDI-MS analysis of patient sera.

One distinct property of these learning-based feature selection methods compared to traditional statistical methods is the coupling of feature selection and sample classification. They implicitly approach the feature selection problem from a multivariate perspective. The significance of a feature depends strongly upon other features. In contrast, the feature selection methods employed in *Dudoit* et al. [34.30] and *Golub* et al. [34.31] are univariate and interactions among genes are ignored.

34.4 Sample Classification

There are many well established discriminant methods, including linear and quadratic discriminant analysis, and k-nearest neighbor, which have been compared in the context of classifying samples using microarray and MS data [34.5, 30]. The majority of these methods were developed in the pregenome era, where the sample size n was usually very large while the number of features p was very small. Therefore, directly applying these methods to genomic and proteomic datasets does not work. Instead, feature selection methods are usually applied to select some "useful" features at first and then the selected features are used to carry out sample classification based on traditional discriminant methods. This two-step approach essentially divides the problem into two separate steps: feature selection and sample classification, unlike the recently developed machine learning methods where the two parts are combined together.

The previously mentioned bagging (*Breiman*) [34.24], boosting (*Freund* and *Schapire*) [34.25], random forest (*Breiman*) [34.32], and support vector machine (*Vapnik*) [34.33] approaches have all been successfully applied to high-dimension genomic and proteomic datasets.

Due to the lack of a genuine testing dataset, cross-validation (CV) has been widely used to estimate the error rate for the classification methods. Inappropriate use of CV may seriously underestimate the real classification error rate. *Ambroise* and *McLachlan* [34.34] discussed the appropriate use of CV to estimate classification error rate, and recommended the use of K-fold cross-validation, e.g. $K = 5$ or 10.

34.5 Random Forest: Joint Modelling of Feature Selection and Classification

Wu et al. [34.5] compared the performance of several classical discriminant methods and some recently developed machine learning methods for analyzing an ovarian cancer MS dataset. In this study, random forest was shown to have good performance in terms of feature selection and sample classification. Here we design an algorithm to get an unbiased estimation of the classification error using random forest and at the same time efficiently extract useful features.

Suppose the preprocessed MS dataset has n samples and p peptides. We use $\{X_k \in \mathbb{R}^p,\ k = 1, 2, \cdots, n\}$ to represent the intensity profile of the kth individ-

ual, and $\{Y_k, k = 1, 2, \cdots, n\}$ to code the sample status.

The general idea of random forest is to combine random feature selection and bootstrap resampling to improve sample classification. We can briefly summarize the general idea as the following algorithm.

General random forest algorithm

1. Specify the number of bootstrap samples B, say 10^5.
2. For $b = 1, 2, \cdots, B$,
   a) Sample with replacement n samples from $\{X_k\}$ and denote the bootstrap samples by $\boldsymbol{X}^b = \{X_{b_1}, \cdots, X_{b_n}\}$, the corresponding response being $\boldsymbol{Y}^b = \{Y_{b_1}, \cdots, Y_{b_n}\}$.
   b) Randomly select m out of p peptides. Denote the selected subset of features by $\{r_1, \cdots, r_m\}$, and the bootstrap samples restricted to this subset by $\boldsymbol{X}_m^b$. Build a tree classifier T_b using $\boldsymbol{Y}^b$ and $\boldsymbol{X}_m^b$. Predict those samples not in the bootstrap samples using T_b.
3. Average the prediction over bootstrap samples to produce the final classification.

For the random forest algorithm from *Breiman* [34.32], randomness is introduced at each node split. Specifically, at each node split, a fixed number of features is randomly selected from all of the features and the best split is chosen among these selected features. For the random subspace method developed by *Ho* [34.35], a fixed number of features is selected at first and is used for the same original data to produce a tree classifier. Thus, both models have the effect of randomly using a fixed subset of features to produce a classifier, but differ in the underlying tree-building method.

Figure 34.6 shows a simple comparison of the two methods. We selected 78 peptides from the ovarian cancer MS data reported by *Wu* et al. [34.5]. Then we apply the two algorithms to numerically evaluate their sample classification performance using the selected subset of features. We want to emphasize that the calculated classification error rate is not a true error rate because we have used the sample status to select 78 peptides first. Our purpose here is just to show a simple numerical comparison of these two methods.

Other important issues in the analysis of MS data include specification of the number of biomarkers and the sample size being incorporated into the experimental design. To estimate the classification error Err, as discussed in *Cortes* et al. [34.36], the inverse power law learning curve relationship between Err and sample size N,

$$\mathrm{Err}(N) = \beta_0 N^{-\alpha} + \beta_1 \,, \tag{34.2}$$

is approximately true for large sample size datasets (usually about tens of thousands of samples); β_1 is the asymptotic classification error and (α, β_0) are positive constants.

Current MS data usually have a relatively small sample size ($N \approx 10^2$) compared to the high-dimension feature space ($p \approx 10^5$). In this situation, it may not be appropriate to rely on the learning curve model to extrapolate β_1, which corresponds to an infinite training sample size $N = \infty$. But within a limited range, this model may be useful to extrapolate the classification error. To estimate parameters $(\alpha, \beta_0, \beta_1)$, we need to obtain at least three observations.

Obviously the classification error Err also depends on the selected number of biomarkers m. We are going to use the inverse-power-law (34.2) to model $\mathrm{Err}(N, m)$.

We proposed the following algorithm to get an unbiased estimate for the classification error rate, which also provides an empirical method to select the number of biomarkers (Wu et al.) [34.37].

CV error estimation using random forest algorithm

1. Specify the number of folders K, say 5, and the range for the number of biomarkers m, say $M = \{20, 21, \cdots, 100\}$. Randomly divide all N samples into balanced K groups.
2. For $k = 1, 2, \cdots, K$ do the following:
   a) Use samples in the kth group as the testing set Ts and all the other samples as the training set Tr.

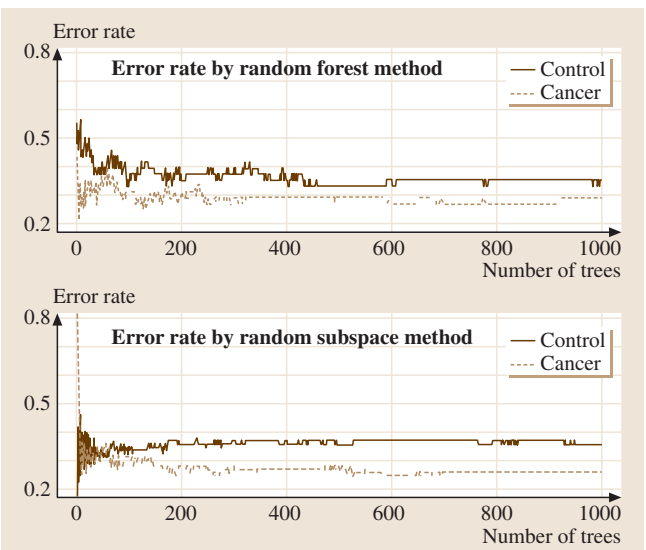


Fig. 34.6 Comparison of the error rates of two random forest algorithms on an ovarian cancer data set. 78 features selected by t-test are used in both algorithms. The two methods give similar performances

b) Apply the random forest algorithm (or any other feature selection method) to the training data Tr. Rank all of the features according to their importance.

c) Use the first $m \in M$ most important features and construct a classifier based on the training set Tr and predict samples in the testing set Ts. We will get a series of error estimates

$$\left\{\epsilon\left(k, m, \frac{K-1}{K}N\right), m \in M\right\} ,$$

where $\frac{K-1}{K}N$ is the effective size of the training set.

d) Use samples in the ith and jth group as the testing set and other $K-2$ groups as the training set. Repeat steps (2.2) and (2.3) to get the error estimate

$$\left\{\epsilon\left(k, m, \frac{K-2}{K}N\right), m \in M\right\} ,$$

where $\frac{K-2}{K}N$ is the effective size of the training set.

e) We can repeat step (2.4) using n of the groups as a testing set and get the error rate

$$\left\{\epsilon\left(k, m, \frac{K-n}{K}N\right), m \in M\right\} ,$$
$$n = 1, 2, \cdots, K-1 .$$

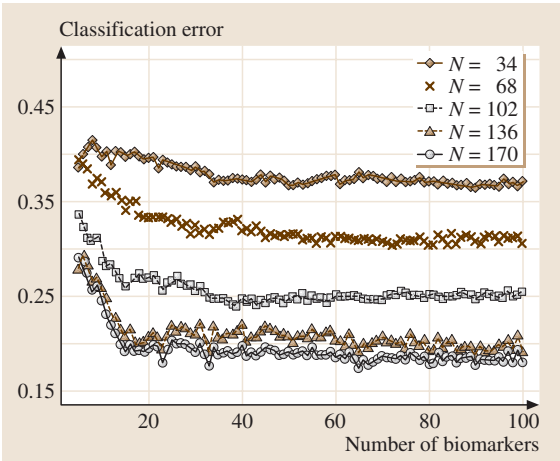


Fig. 34.7 Fivefold cross-validation estimation of classification error rate achieved by applying a random forest algorithm to the ovarian dataset. The error rates for sample size $N = 34, 68, 102, 136$ are obtained from the fivefold CV and the error rate for $N = 170$ is extrapolated from the inverse power law model fitting

3. Average $\epsilon[k, m, N(K-n)/K]$ over K folders to get the final error estimation $\bar{\epsilon}[m, N(K-n)/K]$ for m biomarkers and sample size $N(K-n)/K$.
4. Fit the inverse power law model (34.2) to $\bar{\epsilon}[m, N(K-n)/K]$ for every fixed m and extrapolate the error rate to N samples, $\bar{\epsilon}(m, N)$.

The estimated error rate curve $\bar{\epsilon}(m, n)$ can be used as a guidance for sample size calculation and to select the number of biomarkers.

For K folders, the previous algorithm will involve a total of 2^K training set fittings. If K is relatively large, say 10, the total number of fittings will be very large $(2^K = 1024)$. Note that in the inverse power law model (34.2) we only have three parameters $(\alpha, \beta_0, \beta_1)$. We can carry out just enough training data fitting, say 10, to estimate these three parameters. Then we can use the fitted model to interpolate or extrapolate the classification error rate for other sample size.

Figure 34.7 displays the fivefold CV estimate of the classification error rate achieved by applying the random forest algorithm to the serum mass spectrometry dataset for 170 ovarian cancer patients reported in *Wu* et al. [34.5], where the error rates for the training sample size $N = 34, 68, 102, 136$ are derived from the fivefold CV, and the error rate for $N = 170$ is extrapolated from the inverse power law model fitting.

34.5.1 Remaining Problems in Feature Selection and Sample Classification

As we discussed before, the univariate feature selection based on t-statistics is very sensitive to noise. We can reduce the influence of noise marginally by using additional biological information. But more importantly, we need to develop robust statistical methods. It is conjectured that random forest (*Breiman*) [34.32] does not over-fit. Our experience shows that we can dramatically reduce the classification error rate by incorporating feature selection with the random forest algorithm.

Intuitively, the sample classification error rate will increase with too much noise in the data. In this sense, feature selection will help us to improve the performance of algorithm classification. However, feature selection is usually affected by the small sample size $(n \ll p)$ in genomic and proteomic datasets. If we only select a small number of features, we may miss many "useful" features. One approach would be to couple the fast univariate feature selection with computationally intensive machine learning methods. For example, we can first use univariate feature selection to reduce the number of

features to a manageable size M_0. Then, we can apply the machine learning methods to refine our selection to a small number of target features M_1. Certainly, determining M_0 is a trade-off issue: if M_0 is too small, we will miss many informative features; if M_0 is too large, we will have a heavy computing burden for the following machine learning methods and also make the feature selection unstable.

For the genomic and proteomic data, the large n and small p will make the majority of the traditional statistical methods unusable. Most recently developed machine learning methods are computationally intensive and are often evaluated by empirical performance on some datasets. Statistical methods needed to be developed to bridge the traditional model-based principles and the newly developed machine learning methods.

34.6 Protein/Peptide Identification

34.6.1 Database Searching

MS in combination with database searching has emerged as a key platform in proteomics for high-throughput identification and quantification of proteins. A single stage MS provides a "mass fingerprint" of the peptide products of the enzymatically digested proteins, and this can be used to identify proteins [34.38–45]. This approach is useful for identifying purified proteins (such as proteins in dye-stained spots from 2-D polyacrylamide gels), and may also succeed with simple mixtures containing only 2–3 major proteins. Protein identification based on peptide mass database searching requires both high mass accuracy and that observed peptide masses be matched to a sufficient fraction (e.g. >25%) of the putatively identified protein. The latter task will be made more difficult if the protein has been post-translationally modified at multiple sites. Alternatively, the resulting peptide ions from the first stage MS can be isolated in the mass spectrometer and individually fragmented through CID to produce a tandem MS. In addition to the parent peptide mass, tandem MS provides structural information that can be used to deduce the amino acid sequences of individual peptides. Since tandem MS often identifies proteins using the CID-induced spectrum obtained from a single peptide, this technology is capable of identifying proteins in very complex mixtures such as cell extracts [34.43, 44, 46–61]. In general, database searching methods compare the experimentally observed tandem MS with features predicted for hypothetical spectra from candidate peptides (of equal mass) in the database and then return a ranked listing of the best matches, assuming that the query peptide exists in the protein sequence database. The statistical challenge in MS- and MS/MS-based protein identification is to assess the probability that a putative protein identification is indeed correct. In the case of MS/MS-based approaches, a commonly used criterion is that the observed MS/MS spectra must be matched to at least two different peptides from each identified protein.

34.6.2 De Novo Sequencing

An alternative approach to database searching of uninterpreted tandem MS for peptide identification is De Novo MS/MS sequencing [34.62–65], which attempts to derive a peptide sequence directly from tandem MS data. Although de novo MS/MS sequencing can handle situations where a target sequence is not in the protein database searched, the utility of this approach is highly dependent upon the quality of tandem MS data, such as the number of predicted fragment ion peaks that are observed and the level of noise, as well as the high level of expertise of the mass spectroscopist in interpreting the data, as there is no currently accepted algorithm capable of interpreting MS/MS spectra in terms of a de novo peptide sequence without human intervention. Because of the availability of DNA sequence databases, many of which are genome-level, and the very large numbers of MS/MS spectra (e.g., tens of thousands) generated in a single isotope coded affinity tag or another MS-based protein profiling analysis of a control versus experimental cell extract, highly automated database searching of uninterpreted MS/MS spectra is by necessity the current method of choice for high-throughput protein identification [34.43, 46, 49].

34.6.3 Statistical and Computational Methods

Due to the large number of available methods/alogrithms for MS- and MS/MS-based protein identification, we focus on what we believe are currently the most widely used approaches in the field.

- SEQUEST (Eng et al.) [34.46]

SEQUEST is one of the foremost yet sophisticated algorithms developed for identifying proteins from tandem

MS data. The analysis strategy can be divided into four major steps: data reduction, search for potential peptide matches, scoring peptide candidates and cross-correlation validation. More specifically, it begins with computer reduction of the experimental tandem MS data and only retains the most abundant ions to eliminate noise and to increase computational speed. It then chooses a protein database to search for all possible contiguous sequences of amino acids that match the mass of the peptide with a predetermined mass tolerance. Limited structure modifications may be taken into account as well as the specificity of the proteolytic enzyme used to generate the peptides. After that, SEQUEST compares the predicted fragment ions from each of the peptide sequences retrieved from the database with the observed fragment ions and assigns a score to the retrieved peptide using several criteria such as the number of matching ions and their corresponding intensities, some immonium ions, and the total number of predicted sequence ions. The top 500 best fit sequences are then subjected to a correlation-based analysis to generate a final score and ranking for the sequences.

SEQUEST correlates MS spectra predicted for peptide sequences in a protein database with an observed MS/MS spectrum. The cross-correlation score function provides a measure of the similarity between the predicted and observed fragment ions and a ranked order of relative closeness of fit of predicted fragment ions to other isobaric sequences in the database. However, since the cross-correlation score does not have probabilistic significance, it is not possible to determine the probability that the top-ranked and/or other matches result from random events and are thus false positives. Although lacking a statistical basis, *Eng* et al. [34.46] suggest that a difference greater than 0.1 between the normalized cross-correlation functions of the first- and second-ranked peptides indicates a successful match between the top-ranked peptide sequence and the observed spectrum. A commonly used guideline for Sequest-based protein identification is that observed MS/MS spectra are matched to two or more predicted peptides from the same protein and that each matched peptide meets the 0.1 difference criterion.

- MASCOT (*Perkins* et al.) [34.43]

MASCOT is another commonly used database searching algorithm, which incorporates a probability-based scoring scheme. The basic approach is to calculate the probability (via an approach that is not well described in the literature) that a match between the experimental MS/MS data set and each sequence database entry is a chance event. The match with the lowest probability of resulting from a chance event is reported as the best match. MASCOT considers many key factors, such as the number of missed cleavages, both quantitative and nonquantitative modifications (the number of nonquantitative modifications is limited to four), mass accuracy, the particular ion series to be searched, and peak intensities. Hence, MASCOT iteratively searches for the set with the most intense ion peaks, which provide the highest score – with the latter being reported as $-10\log(P)$, where P is the probability of the match resulting from a random, chance event. *Perkins* et al. [34.43] suggested that the validity of MASCOT probabilities should be tested by repeating the search against a randomized sequence database and by comparing the MASCOT results with those obtained via the use of other search engines.

- Other Methods

In addition to SEQUEST and MASCOT, many other methods have been proposed to identify peptides and proteins from tandem MS data. They range from the development of probability-based scoring schemes, the identification of modified peptides, and checking the identities of peptides and proteins in other miscellaneous fields. Here we give a brief review of these approaches.

Bafna and *Edwards* [34.49] proposed the use of SCOPE to score a peptide with a conditional probability of generating the observed spectrum. SCOPE models the process of tandem MS spectrum generation using a two-step stochastic process. Then SCOPE searches a database for the peptide that maximizes the conditional probability. The SCOPE algorithm works only as well as the probabilities assumed for each predicted fragment of a peptide. Although *Bafna* and *Edwards* [34.49] proposed using a human-curated database of identified spectra to compute empirical estimates of the fragmentation probabilities required by this algorithm, to our knowledge this task has not yet been carried out. Thus, SCOPE is not yet a viable option for most laboratories.

Pevzner et al. [34.48] implemented spectral convolution and spectral alignment approaches to identifying modified peptides without the need for exhaustive generation of all possible mutations and modifications. The advantages of these approaches come with a tradeoff in the accuracy of their scoring functions, and they usually serve as filters to identify a set of "top-hit" peptides for further analysis. *Lu* and *Chen* [34.60] developed a suffix tree approach to reduce search time when identifying modified peptides, but the resulting scores do not have direct probabilistic interpretations.

PeptideProphet [34.53] and ProteinProphet (*Nesvizhskii* et al.) [34.61] were developed at the Institute for

Systems Biology (ISB) to validate peptide and protein identifications using robust statistical models. After scores are derived from the database search, PeptideProphet models the distributions of these scores as a mixture of two distributions, with one consisting of correct matches, and the other consisting of incorrect matches. ProteinProphet takes as input the list of peptides along with probabilities from PeptideProphet, adjusts the probabilities for observed protein grouping information, and then discriminates correct from incorrect protein identifications.

Mann and *Wilm* [34.47] proposed a "peptide sequence tag" approach to extracting a short, unambiguous amino acid sequence from the peak pattern that, when combined with the mass information, infers the composition of the peptide. *Clauser* et al. [34.44] considered the impact of measurement accuracy in protein identification. *Kapp* et al. [34.66] proposed two different statistical methods, the cleavage intensity ratio (CIR) and a linear model, to identify the key factors that influence peptide fragmentation. It has been known for a long time that peptides do not fragment equally and that some bonds are more likely to break than others. However, the chemical mechanisms and physical processes that govern the fragmentation of peptides are highly complex. One can only take results from previous experiments and try to find indicators about such mechanisms. The use of these statistical methods demonstrates that proton mobility is the most important factor. Other important factors include local secondary structure and conformation as well as the position of a residue within a peptide.

34.7 Conclusion and Perspective

While the algorithms for protein identification from tandem MS mentioned above have different emphases, they contain the elements of the following three modules [34.49]:

1. Interpretation [34.67], where the input MS/MS data are interpreted and the output may include parent peptide mass and possibly a partial sequence.
2. Filtering, where the interpreted MS/MS data are used as templates in a database search in order to identify a set of candidate peptides.
3. Scoring, where the candidate peptides are ranked with a score.

Among these three modules, a good scoring scheme is the mainstay. Most database searching algorithms assign a score function by correlating the uninterpreted tandem MS with theoretical/simulated tandem MS for certain peptides derived from protein sequence databases. An emerging issue is the significance of the match between a peptide sequence and tandem MS data. This is especially important in multidimensional LC/MS-based protein profiling where, for instance, our isotope-coded affinity tag studies on crude cell extracts typically identify and quantify two or more peptides from only a few hundred proteins as compared to identifying only a single peptide from a thousand or more proteins. Currently, we require that two or more peptides must be matched to each identified protein. However, if statistically sound criteria could be developed to permit firm protein identifications based on only a single MS/MS spectrum, the useable data would increase significantly. Therefore, it is important and necessary to develop the best possible probability-based scoring schemes, particularly in the case of the automated high-throughput protein analyses used today.

Even for the probability-based algorithms, the efficiencies of score functions can be further improved by incorporating other important factors. For example, statistical models proposed by *Kapp* et al. [34.66] may be used to predict the important factors that govern the fragmentation pattern of peptides and subsequently improve the fragmentation probability as well as the score function in SCOPE [34.49]. In addition, some intensity information can be added to improve score function.

One common drawback of all of these algorithms is the lack of ability to detect modified peptides. Most of the database search methods are not mutation- and modification-tolerant. They are not effective at detecting types and sites of sequence variations, leading to low score functions. A few methods have incorporated mutation and modification into their algorithms, but they can only handle at most two or three possible modifications. Therefore, the identification of modified peptides remains a challenging problem. Theoretically, one can generate a virtual database of all modified peptides for a small set of modifications and match the spectrum against this virtual database. But the size of this virtual database increases exponentially with the number of modifications allowed, making this approach unfeasible. Markov chain Monte Carlo is an appealing approach to identifying mutated and modified peptides. The algorithm may start from a peptide corresponding

to a protein and a "new" candidate peptide with modifications/mutations is proposed according to a set of prior probabilities for different modifications and mutations. The proposed "new" peptide is either rejected or accepted and the procedure can be iterated to sample the posterior distribution for protein modification sites and mutations. However, the computational demands can also be enormous for this approach. Parallel computation and better-constructed databases are necessary to make this approach more feasible.

References

34.1 D. Greenbaum, C. Colangelo, K. Williams, M. Gerstein: Computing protein abundance and mRNA expression levels on a genomic scale, Genome Biol. **4**, 117.1–117.8 (2003)

34.2 M. Wagner, D. Naik, A. Pothen: Protocols for disease classification from mass spectrometry data, Proteomics **3**(9), 1692–1698 (2003)

34.3 Y. Yasui, M. Pepe, M.L. Thompson, B. Adam, G.L. Wright Jr., Y. Qu, J.D. Potter, M. Winget, M. Thornquist, Z. Feng: A data-analytic strategy for protein biomarker discovery: profiling of high-dimensional proteomic data for cancer detection, Biostatistics **4**(3), 449–463 (2003)

34.4 K.R. Coombes, H.A. Fritsche, Jr, C. Clarke, J. Chen, K.A. Baggerly, J.S. Morris, L. Xiao, M. Hung, H.M. Kuerer: Quality control, peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption, ionization, Clinical Chemistry **49**(10), 1615–1623 (2003)

34.5 B. Wu, T. Abbott, D. Fishman, W. McMurray, G. Mor, K. Stone, D. Ward, K. Williams, H. Zhao: Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data, Bioinformatics **19**(13), 1636–1643 (2003)

34.6 Q. Liu, B. Krashnapuram, P. Pratapa, X. Liao, A. Hartemink, L. Carin: Identification of differentially expressed proteins using maldi-tof mass spectra. In: *ASILOMAR Conference: Biological Aspects of Signal Processing* 2003)

34.7 Y. Yasui, D. McLerran, B.L. Adam, M. Winget, M. Thornquist, Z.D.Z.D. Feng: An automated peak identification/calibration procedure for high-dimensional protein measures from mass spectrometers, J. Biomed. Biotec. **4**, 242–248 (2003)

34.8 G.A. Satten, S. Datta, H. Moura, A.R. Woolfitt, G. Carvalho, R. Facklam, J.R. Barr: Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens, Bioinformatics **20**(17), 3128–3136 (2004)

34.9 K.R. Coombes, S. Tsavachidis, J.S. Morris, K.A. Baggerly, M. Hung, H.M. Kuerer: *Improved peak detection, quantification of mass spectrometry data acquired from surface-enhanced laser desorption, ionization by denoising spectra with the undecimated discrete wavelet transform*, Technical report (Univ. Texas M.D. Anderson Cancer Center, Houston 2004)

34.10 T.W. Randolph and Y. Yasui: Multiscale processing of mass spectrometry data, University of Washington Biostatistics Working Paper Series, Number 230, (2004)

34.11 W. Yu, B. Wu, N. Lin, K. Stone, K. Williams, H. Zhao: Detecting, aligning peaks in mass spectrometry data with applications to MALDI, Comput. Biol. Chem. (2005) in press

34.12 R.J.O. Torgrip, M. Aberg, B. Karlberg, S.P. Jacobsson: Peak alignment using reduced set mapping, J. Chemometrics **17**, 573–582 (2003)

34.13 P.H.C. Eilers: Parametric time warping, Analytical Chemistry **76**(2), 404–411 (2004)

34.14 R. Tibshirani, T. Hastie, B. Narasimhan, S. Soltys, G. Shi, A. Koong, Q. Le: Sample classification from protein mass spectrometry, by "peak probability contrasts", Bioinformatics **20**(17), 3034–3044 (2004)

34.15 K.J. Johnson, B.W. Wright, K.H. Jarman, R.E. Synovec: High-speed peak matching algorithm for retention time alignment of gas chromatographic data for chemometric analysis, J. Chromatography A **996**, 141–155 (2003)

34.16 N.V. Nielsen, J.M. Carstensen, J. Smedsgaard: Aligning of single, multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping, J. Chromatography A **805**, 17–35 (1998)

34.17 J. Aach, G.M. Church: Aligning gene expression time series with time warping algorithms, Bioinformatics **17**(6), 495–508 (2001)

34.18 S. Dudoit, Y.H. Yang, T.P. Speed, M.J. Callow: Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments, Stat. Sinica **12**(1), 111–139 (2002)

34.19 V.G. Tusher, R. Tibshirani, G. Chu: Significance analysis of microarrays applied to the ionizing radiation response, Proc. Natl. Acad. Sci. **98**(9), 5116–5121 (2001)

34.20 X. Cui, G.A. Churchill: Statistical tests for differential expression in cDNA microarray experiments, Genome Biology **4**(4), 210 (2003)

34.21 Y. Lai, B. Wu, L. Chen, H. Zhao: Statistical method for identifying differential gene–gene coexpression patterns, Bioinformatics **20**(17), 3146–3155 (2004)

34.22 L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone: *Classification and Regression Trees* (Kluwer Academic, 1984)

34.23 E. C. Gunther, D. J. Stone, R. W. Gerwien, P. Bento, M. P. Heyes: Prediction of clinical drug efficacy by classification of drug-induced genomic expression profiles in vitro, Proc. Natl. Acad. Sci **100**(16), 9608–9613 (2003)

34.24 L. Breiman: Bagging predictors, Machine Learning **24**, 123–140 (1996)

34.25 Y. Freund, R. Schapire: A decision-theoretic generalization of online learning, an application to boosting, J. Computer, System Sci. **55**(1), 119–139 (1997)

34.26 B. Adam, Y. Qu, J. W. Davis, M. D. Ward, M. A. Clements, L. H. Cazares, O. J. Semmes, P. F. Schellhammer, Y. Yasui, Z. Feng: Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men, Cancer Research **62**(13), 3609–3614 (2002)

34.27 M. Dettling, P. Buhlmann: Boosting for tumor classification with gene expression data, Bioinformatics **19**(9), 1061–1069 (2003)

34.28 G. Isabelle, W. Jason, B. Stephen, V. Vladimir: Gene selection for cancer classification using support vector machines, Machine Learning **46**(1–3), 389–422 (2002)

34.29 Y. Qu, B. L. Adam, Y. Yasui, M. D. Ward, L. H. Cazares, P. F. Schellhammer, Z. Feng, O. J. Semmes, G. L. Wright Jr.: Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients, Clin. Chem. **48**(10), 1835–1843 (2002)

34.30 S. Dudoit, J. Fridlyand, T. P. Speed: Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. **97**(457), 77–87 (2002)

34.31 T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, E. S. Lander: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science **286**(5439), 531–537 (1999)

34.32 L. Breiman: Random forests, Machine Learning **45**(1), 5–32 (2001)

34.33 V. N. Vapnik: *Statistical Learning Theory* (Wiley-Interscience, New York 1998)

34.34 C. Ambroise, G. J. McLachlan: Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Acad. Sci. **99**(10), 6562–6566 (2002)

34.35 T. K. Ho: The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. **20**(8), 832–844 (1998)

34.36 C. Cortes, L. D. Jackel, S. A. Solla, V. Vapnik, J. S. Denker: Learning curves: asymptotic values, rate of convergence, Adv. Neural Info. Proc. Systems **6**, 327–334 (1994)

34.37 B. Wu, T. Abbott, D. Fishman, W. McMurray, G. Mor, K. Stone, D. Ward, K. Williams, H. Zhao: Ovarian cancer classification based on mass spectrometry analysis of sera, Cancer Informatics (2005) in press

34.38 W. J. Henzel, T. M. Billeci, J. T. Stults, S. C. Wong, C. Grimley, C. Watanabe: Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases, Proc. Natl. Acad. Sci. **90**, 5011–5015 (1993)

34.39 P. James, M. Quadroni, E. Carafoli, G. Gonnet: Protein identification by mass profile fingerprinting, Biochem. Biophys. Res. Commun. **195**, 58–64 (1993)

34.40 M. Mann, P. Hojrup, P. Roepstorff: Use of mass spectrometric molecular weight information to identify proteins in sequence databases, Biol. Mass Spectrom. **22**, 338–345 (1993)

34.41 D. J. Pappin, P. Hojrup, A. J. Bleasby: Rapid identification of proteins by peptide-mass fingerprinting, Curr. Biol. **3**, 327–332 (1993)

34.42 J. R. Yates III, S. Speicher, P. R. Griffin, T. Hunkapiller: Peptide mass maps: A highly informative approach to protein identification, Anal. Biochem. **214**, 397–408 (1993)

34.43 D. N. Perkins, D. J. Pappin, D. M. Creasy, J. S. Cottrell: Probability-based protein identification by searching sequence databases using mass spectrometry data, J. S. Electrophoresis **20**, 3551–3567 (1999)

34.44 K. R. Clauser, P. Baker, A. I. Burlingame: Role of accurate mass measurement (+/- 10 ppm) in protein identification strategies employing MS or MS/MS and database searching, Anal. Chem. **71**, 2871–2882 (1999)

34.45 W. Zhang, B. T. Chait: ProFound: An expert system for protein identification using mass spectrometric peptide mapping information, Anal. Chem. **72**, 2482–2489 (2000)

34.46 J. K. Eng, A. L. McCormack, J. R. Yates: An approach to correlate MS/MS data to amino acid sequences in a protein database, J. Am. Soc. Mass Spectrom. **5**, 976–989 (1994)

34.47 M. Mann, M. S. Wilm: Error-tolerant identification of peptides in sequence databases by peptide sequence tags, Anal. Chem. **66**, 4390–4399 (1994)

34.48 P. A. Pevzner, V. Dancik, C. L. Tang: Mutation-tolerant protein identification by mass spectrometry, J. Comput. Biol. **7**, 777–787 (2000)

34.49 V. Bafna, N. Edwards: SCOPE: A probabilistic model for scoring tandem mass spectra against a peptide database, Bioinformatics **17**, S13–21 (2001)

34.50 B. T. Hansen, J. A. Jones, D. E. Mason, D. C. Liebler: SALSA: A pattern recognition algorithm to detect electrophile-adducted peptides by automated

evaluation of CID spectra in LC-MS-MS analyses, Anal. Chem. **73**, 1676–1683 (2001)

34.51 D. M. Creasy, J. S. Cottrell: Error-tolerant searching of uninterpreted tandem mass spectrometry data, Proteomics **2**, 1426–1434 (2002)

34.52 H. I. Field, D. Fenyo, R. C. Beavis: RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in arelational database, Proteomics **2**, 36–47 (2002)

34.53 A. Keller, A. I. Nesvizhskii, E. Kolker, R. Aebersold: Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search, Anal. Chem. **74**, 5389–5392 (2002)

34.54 M. J. MacCoss, C. C. Wu, J. R. Yates: Probability-based validation of protein identifications using amodified SEQUEST algorithm, Anal. Chem. **74**, 5593–5599 (2002)

34.55 D. C. Anderson, W. Li, D. G. Payan, W. S. Noble: A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores, J. Proteome Res. **2**, 137–146 (2003)

34.56 J. Colinge, A. Masselot, M. Giron, T. Dessigny, J. Magnin: OLAV: towards high throughput tandem mass spectrometry data identification, Proteomics **3**, 1454–1463 (2003)

34.57 E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, A. Bairoch: ExPASy: The proteomics server for in-depth protein knowledge and analysis, Nucleic Acids Res. **3**, 3784–3788 (2003)

34.58 M. Havilio, Y. Haddad, Z. Smilansky: Intensity-based statistical scorer for tandem mass spectrometry, Anal. Chem. **75**, 435–444 (2003)

34.59 P. Hernandez, R. Gras, J. Frey, R. D. Appel: Popitam: towards new heuristic strategies to improve protein identification from tandem mass spectrometry data, Proteomics **3**, 870–878 (2003)

34.60 B. Lu, T. Chen: A suffix tree approach to the interpretation of tandem mass spectra: applications to peptides of non-specific digestion, post-translational modifications, Bioinformatics **19**, 113–121 (2003)

34.61 A. I. Nesvizhskii, A. Keller, E. Kolker, R. Aebersold: A statistical model for identifying proteins by tandem mass spectrometry, Anal. Chem. **75**, 4646–4658 (2003)

34.62 J. A. Taylor, R. S. Johnson: Sequence database searches via de novo peptide sequencing by tandem mass spectrometry, Rapid Commun. Mass Spectrom. **11**, 1067–75 (1997)

34.63 V. Dancik, T. A. Addona, K. R. Clauser, J. E. Vath, P. A. Pevzner: De Novo peptide sequencing via tandem mass spectrometry, J. Comput. Biol. **6**, 327–342 (1999)

34.64 T. Chen, M. Y. Kao, M. Tepel, J. Rush, G. M. Church: A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry, J. Comput. Biol. **8**, 325–337 (2001)

34.65 B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, G. Lajoie: PEAKS: Powerful software for peptide de novo sequencing by tandem mass spectrometry, Rapid Commun. Mass Spectrom. **17**, 2337–2342 (2003)

34.66 E. A. Kapp, F. Schütz, G. E. Reid, J. S. Eddes, R. L. Moritz, R. A. J. O'Hair, T. P. Speed, R. J. Simpson: Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation, Anal. Chem. **75**, 6251–6264 (2003)

34.67 D. C. Chamrad, G. Koerting, J. Gobom, H. Thiele, J. Klose, H. E. Meyer, M. Blueggel: Interpretation of mass spectrometry data for high-throughput proteomics, Anal. Bioanal. Chem. **376**, 1014–1022 (2003)

35. Radial Basis Functions for Data Mining

This chapter deals with the design and applications of the radial basis function (RBF) model. It is organized into three parts. The first part, consisting of Sect. 35.1, describes the two data mining activities addressed here: classification and regression. Next, we discuss the important issue of bias–variance tradeoff and its relationship to model complexity. The second part consists of Sects. 35.2 to 35.4. Section 35.2 describes the RBF model architecture and its parameters. In Sect. 35.3.1 we briefly describe the four common algorithms used for its design: clustering, orthogonal least squares, regularization, and gradient descent. In Sect. 35.3.2 we discuss an algebraic algorithm, the SG algorithm, which provides a step-by-step approach to RBF design. Section 35.4 presents a detailed example to illustrate the use of the SG algorithm on a small data set. The third part consists of Sects. 35.5 and 35.6. In Sect. 35.5 we describe the development of RBF classifiers for a well-known benchmark problem to determine whether Pima Indians have diabetes. We describe the need for and importance of partitioning the data into training, validation, and test sets. The training set is employed to develop candidate models, the validation set is used to select a model, and the generalization performance of the selected model is assessed using the test set. Section 35.6 describes a recent data mining application in bioinformatics, where the objective is to analyze the gene expression profiles of Leukemia data from patients whose classes are known to predict the target cancer class. Finally, Sect. 35.7 provides concluding remarks and directs the reader to related literature. Although the material in this chapter is applicable to other types of basis funktions, we have used only the Gaussian function for illustrations and case studies because of its popularity and good mathematical properties.

Neural networks have been used extensively to model unknown functional relationships between input and output data. The radial basis function RBF model is a special type of neural network consisting of three layers: input, hidden, and output. It represents two sequential mappings. The first nonlinearly maps the input data via basis functions in the hidden layer. The second, a weighted mapping of the basis function outputs, generates the model output. The two mappings are usually treated separately, which makes RBF a very versatile modeling technique. There has been some debate about whether RBF is biologically plausible, and hence whether it really is a neural network model. Nevertheless, it has become an established model for diverse classification and regression problems. For example, it has been successfully employed in areas such as data mining, medical diagnosis, face and speech recognition, robotics, forecasting stock prices, cataloging objects in the sky, and bioinformatics.

RBF networks have their theoretical roots in regularization theory and were originally developed by Russian mathematicians in the 1960s. They were used for strict interpolation among multidimensional data [35.1], where it is required that every input be mapped to a corresponding output. *Broomhead* and *Lowe* [35.2] used the RBF model for approximation. The relation-

ship between the use of RBF for strict interpolation and approximation is of special interest in this chapter. Further, they have been shown to possess very important mathematical properties of universal and best approximation [35.3]. A function approximation scheme is said to have the property of universal approximation if the set of functions supported by the approximation scheme is dense in the space of the continuous functions defined on the input domain, and it has the property of best approximation if there is one function among this set that has the lowest approximating error for any given function to be approximated. This provides a strong mathematical justification for their practical application, since the popular multilayer perceptrons approach, for example, does not possess the property of best approximation.

In this chapter we describe the radial basis network architecture, its design and applications. The data mining problem we address is described in Sect. 35.1. The RBF model and its parameters are described in Sect. 35.2. Sect. 35.3 presents some common design algorithms. An important design problem relates to determining the number of basis functions in the hidden layer and their parameters. A new algebraic algorithm, the SG algorithm, provides a systematic methodology for doing so and is also discussed in Sect. 35.3. An illustrative modeling example is described in Sect. 35.4, and a benchmark case study about diabetes classification is presented in Sect. 35.5. An important data mining application for cancer class prediction based on microarray data analysis is described in Sect. 35.6. Finally, some concluding remarks are presented in Sect. 35.7.

35.1 Problem Statement

Knowledge discovery applications are aimed at extracting accurate, previously unknown, useful, and actionable information from databases, and the related discipline is known as knowledge discovery in databases KDD. The usual steps in this process, often followed iteratively, are: selection of the target data from the raw databases; its processing and transformation; information extraction (called "data mining") using algorithmic approaches; interpretation of the information gained; and its useful application. Data mining is the key phase in this process and is main interest in this chapter. For a detailed description of this discipline, see [35.4–6].

The data available is a collection of records, each record itself being a collection of fields or data items. This tabular data is the input to a data mining algorithm, the output of which is the desired information or the knowledge sought. Usual data mining applications include data characterization, pattern recognition, rule extraction, clustering, trend analysis, and visualization. In this chapter, we address only pattern recognition. The pattern recognition task can be described as the construction of a model for input-output mapping on the basis of available tabular data. Such data are called the training sample. The inputs are d-dimensional independent variables or features ($\boldsymbol{x}$'s), and the output is a one-dimensional dependent variable (y). The two common pattern recognition tasks are classification and regression. In the case of classification, y represents one of a possible L classes. Most applications, however, are binary classification problems; in other words $L = 2$ in most practical cases. In regression problems y is a continuous variable. The constructed model is employed to predict the output y for future observed input $\boldsymbol{x}$'s. The objective is to seek a data mining algorithm that predicts y as accurately as possible. In other words, we seek to minimize the prediction error on future data.

In classification problems, a commonly used prediction error measure is the "classification error" (CE), which is defined as the ratio of misclassified objects to the total number of objects. For regression problems, the mean squared error (MSE) is generally employed. It is the averaged sum of squared discrepancies between the actual and the predicted values. The model performance is computed for the training data. An independent data set that is representative of the data used for training and is called the "validation set" is employed to compare the performance of the derived models. Then, yet another independent data set, called the "test set", is employed to assess the test error of the selected model as a performance measure on future data, for which the model was developed in the first place.

In the design and selection of RBF models, we prefer a parsimonious model; in other words, one with the smallest number of terms that provides the desired performance. However, it is well known that a model with too few terms can suffer from underfitting, while one with too many can result in overfitting and will therefore fail to generalize well on future data. This problem

is also known as the "bias-variance dilemma" in machine learning and statistics literature [35.5,7,8]. Simple models tend to have high bias and low variance, while complex models tend to have low bias and high variance. A graphical illustration of this phenomenon is shown in Fig. 35.1. The objective of modeling is to seek a compromise or tradeoff between bias and variance, and to find a model of just the right complexity. For the RBF model, as we discuss below, this means that we seek a model with just enough basis functions in the hidden layer. In other words, we seek a model that has a low training error and a low generalization error as assessed by the error on test data. In Fig. 35.1 this idealized situation is labeled as the "best model".

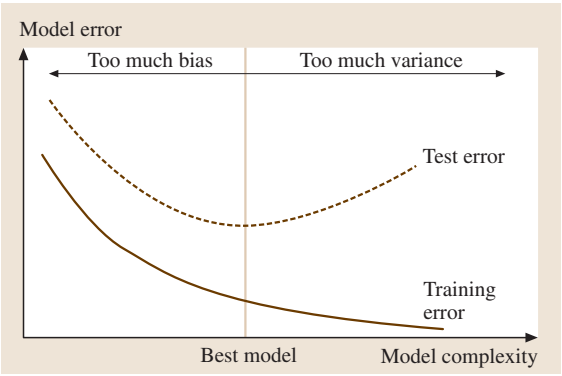


Fig. 35.1 Typical behavior of training error and test error

35.2 RBF Model and Parameters

A typical RBF network is shown in Fig. 35.2. It has three layers: input, hidden and output. The input layer consists of an $n \times d$ input data matrix $\mathbf{X}$:

$$\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)^T \in \mathbb{R}^{n \times d} , \tag{35.1}$$

where $\boldsymbol{x}_i$, i= 1, 2, . . . , n are the d-dimensional vectors and n is the size of the data. The hidden layer has m basis functions $\phi_1(\cdot)$, $\phi_2(\cdot)$, . . . , $\phi_m(\cdot)$ centered at basis function centers $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_m$, respectively, and connected to the output layer with weights $w_1, w_2, \ldots, w_m$, respectively. The basis functions transform the input data matrix via nonlinear mappings based on using the Euclidean distance between the input vector $\boldsymbol{x}$ and the prototype vectors $\boldsymbol{\mu}_j, j = 1, \ldots,$ m. This mapping can be represented as follows, where $\|\cdot\|$ is the Euclidean norm:

$$\phi_j(\boldsymbol{x}) = \phi\left(\left\|\boldsymbol{x} - \boldsymbol{\mu}_j\right\|\right), j = 1, 2, ..., m. \tag{35.2}$$

The $n \times d$ input matrix is thus transformed by the m basis functions into the following $n \times m$ design matrix $\boldsymbol{\Phi}$. In this matrix, the jth column represents the outputs from the jth basis function, $j = 1, 2, \ldots,$m.

$$\boldsymbol{\Phi} = \begin{array}{c} \begin{matrix} \Phi_1 & \hbar & \Phi_j & \hbar & \Phi_m \end{matrix} \\ \begin{pmatrix} \phi_1(x_1) & \hbar & \phi_j(x_1) & \hbar & \phi_m(x_1) \\ \phi_1(x_2) & \hbar & \phi_j(x_2) & \hbar & \phi_m(x_2) \\ \hbar & \hbar & \hbar & \hbar & \hbar \\ \phi_1(x_n) & \hbar & \phi_j(x_n) & \hbar & \phi_m(x_n) \end{pmatrix} \end{array} . \tag{35.3}$$

Several types of basis function have been considered in the literature. The common ones are Gaussian, thin plate spline, inverse multiquadratic, and cubic [35.7,9]. However, the basis function most commonly used for most applications is the Gaussian. Its form is $\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$, where σ is a parameter that controls the smoothness properties of the approximating function. The expression for the jth Gaussian function mapping can be explicitly written as

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\left\|\boldsymbol{x} - \boldsymbol{\mu}_j\right\|^2}{2\sigma_j^2}\right) , \tag{35.4}$$

where $\boldsymbol{\mu}_j$ is the center and σ_j is the width of the jth basis function, $j = 1, 2, \ldots,$m. On substituting in (35.3), we

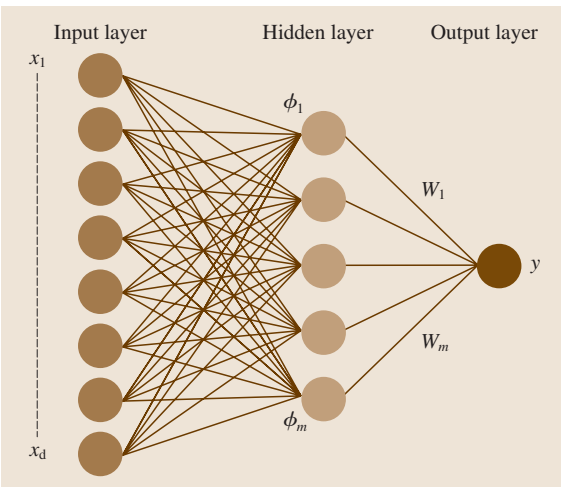


Fig. 35.2 Radial basis function network

can write the expression for a Gaussian design matrix, $\boldsymbol{\Phi}$, as

$$\boldsymbol{\Phi} = \begin{array}{c} \begin{array}{ccccc} \Phi_1 & \hbar & \Phi_j & \hbar & \Phi_m \end{array} \\ \begin{pmatrix} \exp\left(-\frac{\|\boldsymbol{x}_1-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}\right) & \hbar & \exp\left(-\frac{\|\boldsymbol{x}_1-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right) & \hbar & \exp\left(-\frac{\|\boldsymbol{x}_1-\boldsymbol{\mu}_m\|^2}{2\sigma_m^2}\right) \\ \exp\left(-\frac{\|\boldsymbol{x}_2-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}\right) & \hbar & \exp\left(-\frac{\|\boldsymbol{x}_2-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right) & \hbar & \exp\left(-\frac{\|\boldsymbol{x}_2-\boldsymbol{\mu}_m\|^2}{2\sigma_m^2}\right) \\ \hbar & \hbar & \hbar & \hbar & \hbar \\ \exp\left(-\frac{\|\boldsymbol{x}_n-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}\right) & \hbar & \exp\left(-\frac{\|\boldsymbol{x}_n-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right) & \hbar & \exp\left(-\frac{\|\boldsymbol{x}_n-\boldsymbol{\mu}_m\|^2}{2\sigma_m^2}\right) \end{pmatrix} \end{array} \tag{35.5}$$

For the special case where the number of basis functions equals the number of data vectors (when $m = n$), and if we use the n input data vectors as the basis function centers, the matrix in (35.5) is called the interpolation matrix. It is this matrix that is employed for the strict interpolation problem mentioned earlier and is of special interest in the SG algorithm. To compute the output, the entries in the matrix given by (35.5) are combined linearly according to the weights. The resulting values at the output node are then given by

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} w_j \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right), \tag{35.6}$$

where $f(\boldsymbol{x})$ represents the Gaussian RBF output for the input vector $\boldsymbol{x}$. Thus, for n input vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$, the output layer consists of n outputs, one for each $\boldsymbol{x}$, as indicated below,

$$\left[f(\boldsymbol{x}_1),\, f(\boldsymbol{x}_2),\, \ldots,\, f(\boldsymbol{x}_n)\right]^{\mathrm{T}} = \boldsymbol{\Phi}\boldsymbol{w}\,. \tag{35.7}$$

Thus, we see that a Gaussian RBF model is fully defined by the number of basis functions (m), their centers $[\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_m)]$, widths $[\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_m)]$, and the weights $[\boldsymbol{w} = (w_1, w_2, \ldots, w_m)]$ to the output layer. In most applications, and in this chapter, a global width σ is used for each basis function. The parameters m, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ define the hidden layer (the nonlinearity of the RBF model). The weights ($\boldsymbol{w}$) define the linear part as indicated in Fig. 35.2. This completes discussion of the radial basis function model and its parameters. We now move on to discuss RBF model development; in other words, the determination of its parameters from the training sample or the given input-output data set.

35.3 Design Algorithms

A common characteristic of most design or training algorithms used for RBF models is that they employ a two-stage training procedure. In the first stage, only the input data is used to determine the basis function parameters. For the Gaussian case, these are the number of basis functions, their centers, and their widths. Once the basis function parameters are determined, the weights are found in the second stage to minimize some error measure. There are a large number of procedures available in the literature for RBF design. We first describe the four commonly used ones and then a relatively new algorithm called the SG algorithm.

35.3.1 Common Algorithms

Clustering: [35.10] A set of centers for basis functions can be obtained by employing clustering techniques on the input data. The k-means clustering algorithm [35.5] is used to locate a set of k basis function centers. For a specified k, the algorithm seeks to partition the input data into k disjoint subsets, each of which corresponds to a cluster. Once the cluster membership is determined, the averages of the data points in these clusters are chosen as the centers of the k basis functions in the RBF model, and m is taken to equal k. Next, the widths of the basis functions are determined by a P-nearest neighbor heuristic [35.5]. Thus, if P=1, the width of the jth basis function is set to be the Euclidean distance between its own center and the center of its nearest neighbor.

Orthogonal Least Square: [35.11] In this procedure a set of vectors is constructed in the space spanned by the vectors of hidden unit outputs for the training set and then by directly finding the center of an additional basis function such that it gives the greatest reduction in residual sum-of-square error. The stopping criterion employed is a threshold on the fraction of the variance explained by the model.

Regularization: [35.7] These procedures are motivated by the theory of regularization. A regularization parameter is used to control the smoothness properties of a mapping function by adding an extra term to the minimized error function that is designed to penalize mappings that are not smooth.

Gradient Descent: [35.10] Such training algorithms are fully supervised gradient-descent methods over some error measure. Specifically, the model parameters are updated as a function of this error measure according to some specified learning rates associated with the RBF parameters.

35.3.2 SG Algorithm

In the SG algorithm, the parameters $(m, \boldsymbol{\sigma}, \boldsymbol{\mu})$ of the nonlinear mapping are first determined from the input matrix $\mathbf{X}$ without referencing the output values. Then, the linear parameters $(\boldsymbol{w})$ are determined by referencing the output y's. The SG algorithm consists of four steps, given below. These steps are shown schematically in Fig. 35.3.

- Step 1: Select a range of values for global width, $\boldsymbol{\sigma}$, and a representation capability measure, δ, according to the heuristics given below.
- Step 2: Determine a value of m that satisfies the δ criterion. This step involves singular value decomposition of the interpolation matrix computed from the input data matrix $\mathbf{X}$ for a chosen $\boldsymbol{X}\sigma$.
- Step 3: Determine centers for the m basis functions that maximize structural stability provided by the selected model complexity, m. This step involves the use of QR factorization.
- Step 4: Compute weights using the pseudoinverse and estimate the output values.

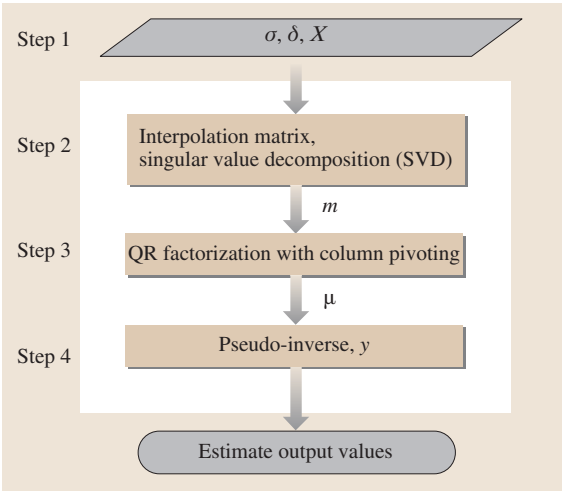


Fig. 35.3 The four steps of the SG RBF modeling algorithm

Note that the choice of parameters in Step 1 affects the quality of the developed model. Heuristically, we take $\boldsymbol{\sigma}$ to be approximately in the range of 0 to $0.75\sqrt{d/2}$, and δ is taken to be in the range 0.1% to 1.0% [35.12], where d is the dimensionality of the input data points. RBF models are then developed for a few judicially chosen values of these parameters. The performance of these models is assessed using some prespecified approach, and the most appropriate model is selected. This process is illustrated in Sects. 35.5 and 35.6 for two real-world data sets.

35.4 Illustrative Example

In this section we illustrate the use of the SG algorithm on a small data set generated from the following sine function [35.7]:

$$h(x) = 0.5 + 0.4\sin(2\pi x)\,.$$

Five values of the above function are computed at equal intervals of x in the range 0.0 to 1.0. Then random noise values, generated from a Gaussian distribution with mean zero and variance 0.25, are added to $h(x)$ to obtain five data points. The data set is listed in Table 35.1 along with the true $h(x)$ values. Our objective is to seek a good approximation for the unknown function $h(x)$, based only on the x and observed y data. A plot of the true $h(x)$ and the observed y values is shown in Fig. 35.4. Also shown is an approximated or estimated function found using interpolation, as discussed next.

Table 35.1 Dataset for illustrative example

i	x_i	$h(x_i)$	y_i
1	0.00	0.50	0.5582
2	0.25	0.90	0.9313
3	0.50	0.50	0.5038
4	0.75	0.10	0.1176
5	1.00	0.50	0.4632

First, we consider the strict interpolation problem for this data set, then we illustrate the use of the SG algorithm for approximation. The interpolation problem is to determine a Gaussian RBF that gives exact outputs for each x; in other words, we seek a model whose output is exactly equal to the y value corresponding to that x given in Table 35.1. For this we construct an inter-

polation matrix with five basis functions, one centered at each input value x. Suppose we use a global width $\sigma = 0.4$. Then the five Gaussian basis functions, each with $\sigma = 0.4$, will be centered at the five x values of Table 35.1 and will map the input data into an interpolation matrix according to the expressions in (35.5) with $m = n = 5$. For example, the column $\boldsymbol{\Phi}_2$ in matrix $\boldsymbol{\Phi}$ is obtained according to (35.3) and (35.4) by substituting $\boldsymbol{\mu} = 0.25$ and the five x values given below.

$$\Phi_2(x) = \begin{pmatrix} \phi_2(x_1) \\ \phi_2(x_2) \\ \phi_2(x_3) \\ \phi_2(x_4) \\ \phi_2(x_5) \end{pmatrix} = \begin{pmatrix} \exp\left(-\frac{\|0.00-0.25\|^2}{2(0.4)^2}\right) \\ \exp\left(-\frac{\|0.25-0.25\|^2}{2(0.4)^2}\right) \\ \exp\left(-\frac{\|0.50-0.25\|^2}{2(0.4)^2}\right) \\ \exp\left(-\frac{\|0.75-0.25\|^2}{2(0.4)^2}\right) \\ \exp\left(-\frac{\|1.00-0.25\|^2}{2(0.4)^2}\right) \end{pmatrix}$$

$$= \begin{pmatrix} 0.8226 \\ 1.0000 \\ 0.8226 \\ 0.4578 \\ 0.1724 \end{pmatrix} .$$

Other columns of the matrix are similarly computed for different $\boldsymbol{\mu}$'s. The final 5×5 interpolation matrix is given below.

$$\boldsymbol{\Phi} = \begin{array}{ccccc} \Phi_1 & \Phi_2 & \Phi_3 & \Phi_4 & \Phi_5 \\ 1.0000 & 0.8226 & 0.4578 & 0.1724 & 0.0439 \\ 0.8226 & 1.0000 & 0.8226 & 0.4578 & 0.1724 \\ 0.4578 & 0.8226 & 1.0000 & 0.8226 & 0.4578 \\ 0.1724 & 0.4578 & 0.8226 & 1.0000 & 0.8226 \\ 0.0439 & 0.1724 & 0.4578 & 0.8226 & 1.0000 \end{array} \quad (35.8)$$

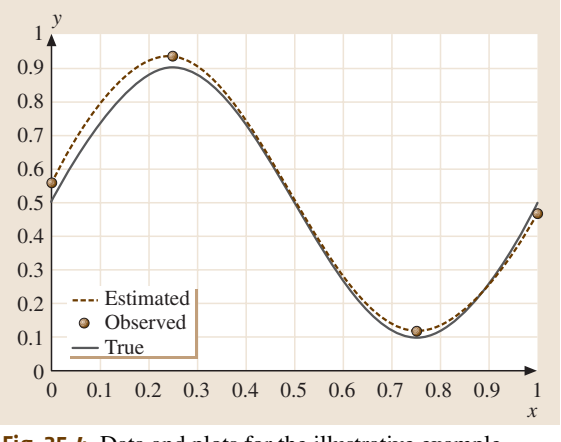


Fig. 35.4 Data and plots for the illustrative example

This matrix is symmetric because we have chosen a global width for the basis functions. Also, its diagonal values are 1.0 because the x values are listed in increasing order and because the height of the basis function at its center is always 1.0. The above interpolation matrix and the y_i vector in Table 35.1 are used to compute weights $\boldsymbol{w} = (w_1, w_2, \ldots, w_5)^T$ of the links to the output node using the pseudoinverse. Finally, the interpolated function is obtained from (35.6) and (35.7) as

$$\begin{aligned} \hat{y} = & -2.09 \exp\left(-\frac{\|x-0.0\|^2}{2(0.4)^2}\right) \\ & +3.87 \exp\left(-\frac{\|x-0.25\|^2}{2(0.4)^2}\right) \\ & -0.06 \exp\left(-\frac{\|x-0.50\|^2}{2(0.4)^2}\right) \\ & +3.63 \exp\left(-\frac{\|x-0.75\|^2}{2(0.4)^2}\right) \\ & +2.91 \exp\left(-\frac{\|x-1.0\|^2}{2(0.4)^2}\right) . \end{aligned}$$

Note that here $\hat{y}$ is the weighted sum of the basis function outputs and consists of five terms, one corresponding to each basis function. A plot of this function is shown in Fig. 35.4, where the estimated values are exactly equal to the observed y's since we are dealing with exact interpolation.

In practical problems, exact interpolation is undesirable because it represents extreme overfitting. Referring to Fig. 35.1, this indicates a very complex model that will not perform well on new data in the sense that it will exhibit a high generalization error. In practice, we seek an approximate model according to the guidelines discussed in Sect. 35.1. To achieve this goal we use the SG algorithm, where the user controls the tradeoff between underfitting and overfitting or between bias and variance by specifying the values of δ [35.12]. As indicated above, for practical applications, $\delta = 0.1\%$ to 1% seems to be a good set of values to consider. The RBF model is then designed according to the above four-step procedure. We provide a description of this procedure for the sine data below. However, details of the singular value decomposition and QR factorization are beyond the scope of this chapter and can be found in [35.12, 13].

The starting point in the SG algorithm is the selection of σ and δ. For the sine data, suppose

$\sigma = 0.4$ and $\delta = 0.01$. The 5×5 interpolation matrix for this σ is computed as shown above. In step 2, its singular value decomposition yields five singular values. Using these and $\delta = 0.01$, for this example data, we obtain $m = 4$. Then, in step 3, QR factorization identifies the four centers for the basis functions as being $\mu_1 = 1.00$, $\mu_2 = 0.00$, $\mu_3 = 0.75$ and $\mu_4 = 0.25$. Finally, the weights are obtained in step 4 as $w_1 = -2.08$, $w_2 = 3.82$, $w_3 = -3.68$ and $w_4 = 2.92$. Thus, an approximation of the unknown function $h(x)$ based on the available data x and y in Table 35.1 is provided by an RBF model

$$\hat{y} = \sum_{j=1}^{4} w_j \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right),$$

where w_j and μ_j are as listed above and $\sigma_j = 0.4$ for $j = 1, 2, 3, 4$. This estimate of y is based on the four basis functions selected by the SG algorithm for $\delta = 0.01$. Note that in this simple example we obtained a value of m that is almost the same as n. However, in practical applications with real-world data, the design value of m is generally much smaller than the n as seen in Sects. 35.5 and 35.6.

35.5 Diabetes Disease Classification

This benchmark problem is taken from the Proben1 data set of the UCI repository [35.14]. It was studied in Lim [35.15] using the SG algorithm. The objective is to develop a classification model to determine whether diabetes of Pima Indians is positive or negative, based on personal data such as age, number of times pregnant, and so on. Other factors considered include the results from medical examinations, such as data on blood pressure, body mass index, and results of glucose tolerance tests. There are eight inputs, two outputs, 768 examples, and no missing values in this data set. A summary of the input and output attributes and the encoding scheme employed for data processing is given in Table 35.2. Here, all inputs are continuous and each is normalized to a range of 0 to 1 for data preprocessing. Attribute number 9 is the output, consisting of two values, diabetes or no diabetes. Before proceeding with RBF model development we need to decide upon an approach to model evaluation. This point is discussed next.

The generalization performance of an RBF model relates to its predictive ability on some future data. Therefore, we need to be able to assess this performance during the model building process. For applications where we have adequate data, the best approach is to randomly divide the available data into three sets: training set, validation set, and test set [35.5]. We use the training set for model development, the validation set to compare the developed models and, usually, select the model with the smallest error on the validation set. The test set is not used until after the final model is selected. The performance of the selected model on the test set is used as a measure of its generalization performance. A common practice is to split the data into 50% for training and 25% each for validation and test sets. Using this

Table 35.2 Data description for the diabetes example

Inputs (8)			
Attribute No.	**No. of attributes**	**Attribute meaning**	**Values and encoding**
1	1	Number of times pregnant	0 ... 17 → 0 ... 1
2	1	Plasma glucose concentration after 2 h in an oral glucose tolerance test	0 ... 199 → 0 ... 1
3	1	Diastolic blood pressure (mm Hg)	0 ... 122 → 0 ... 1
4	1	Triceps skin fold thickness (mm)	0 ... 99 → 0 ... 1
5	1	2-hour serum insulin (mu U/ml)	0 ... 846 → 0 ... 1
6	1	Body mass index (weight in kg/(height in m)^2)	0 ... 67.1 → 0 ... 1
7	1	Diabetes pedigree function	0.078 ... 2.42 → 0 ... 1
8	1	Age (years)	21 ... 81 → 0 ... 1
Output (2)			
9	2	No diabetes Diabetes	1 0 0 1

Table 35.3 RBF models for the diabetes example

$\delta = 0.01$ Model	m	σ	Classification error (CE) (%)		
			Training	Validation	Test
A	18	0.6	20.32	23.44	24.48
B	9	0.7	21.88	21.88	22.92
C	9	0.8	22.66	21.35	23.44
D	8	0.9	22.92	21.88	25.52
E	8	1.0	23.44	21.88	25.52
F	7	1.1	26.04	30.21	30.21
G	6	1.2	25.78	28.13	28.13
H	5	1.3	25.26	31.25	30.73

split for this application, we divide the set of 768 patients into 384 for training, 192 for validation and 192 for the test set.

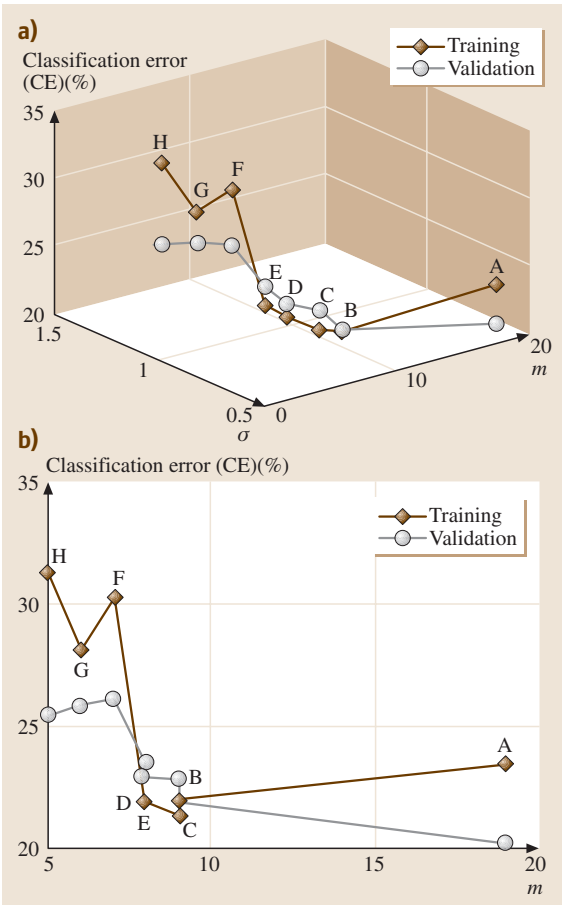


Fig. 35.5 Plots of training and validation errors for the diabetes example ($\delta = 0.01$)

We employ the heuristics given in Sect. 35.3 and select values of width (σ) in the range 0.6 to 1.3. Also, we use $\delta = 0.01$, 0.005 and 0.001. The algorithm is then executed to develop RBF models. For each model, the training and validation classification errors(CE) are also computed. However, to provide a better insight, the test error is also computed here. The classification results for eight models (A to H) for $\delta = 0.01$ are shown in Table 35.3. We note that as σ decreases from 1.3 to 0.6, the m value increases, and the training CE decreases from 25.26% to 20.32%. However, the validation CE first decreases from 31.25% to 21.35% and then increases to 23.44%. The test error first decreases with increasing m and then begins to increase. The validation errors, though used for different purposes, tend to exhibit similar behavior with respect to m. The pattern of CE error behavior is shown graphically in Fig. 35.5. Here, the errors are shown with respect to m and σ in Fig. 35.5a and as a function of m alone in Fig. 35.5b. We note that the training and validation error behavior is quite similar to the theoretical pattern discussed above and that depicted in Fig. 35.1. However, for some models the validation error is smaller than the training error. Also, the error behavior is not monotonic. This can and does happen in practical applications due to random variations in the real-world data assigned to the training, validation, and test sets.

To select the model, we evaluate the validation CE for models A to H. The minimum value occurs for model C, and hence this model is selected as the preferred model. The test CE for this model is 23.44. Next, models were developed for $\delta = 0.005$ and 0.001. However, the details of these models are not shown here. The best models for these two cases and for $\delta = 0.01$ are listed in Table 35.4.

To select the final model, we consult the results in Table 35.4 and note that the RBF model with the smallest

Table 35.4 Selected models and error values for the diabetes example

Classification error (CE) (%)					
δ	m	σ	Training	Validation	Test
0.01	9	0.8	22.66	21.35	23.44
0.005	9	0.8	22.66	21.35	23.44
0.001	10	1.2	22.66	20.83	23.96

validation error is the model with $m = 10, \sigma = 1.2$. This is our final choice for modeling the classification of diabetes. Its test error is 23.96. What this says is that when this model is used to evaluate future patients for diabetes or no diabetes, the model will misclassify, on average, about 24% of the patients. Also, note that the design value of m, the number of basis functions, is only 10, while $n = 384$ for the training data. Thus, we see that a much simpler model than strict interpolation requires provides a good classifier. If we were to use $m = n = 384$ in this case, all patients in the training set would be correctly classified with $CE = 0.0$. However, the n performance of such a model on the validation and test sets is likely to be very poor.

35.6 Analysis of Gene Expression Data

Now we describe a data mining application of the RBF model to binary cancer classification based on gene expression data from DNA microarray hybridization experiments [35.16]. Cancer class prediction is crucial to its treatment, and developing an automated analytical approach for classification based upon the microarray expression is an important task [35.16]. A generic approach to classifying two types of acute leukemia based on the monitoring of gene expression by DNA microarrays was originally pioneered by [35.17]. We employ their data set to illustrate the classifier development process and use sensitivity analyses in order to select an appropriate cancer classification model. The goal is to construct a classifier that distinguishes between two cancer classes based on gene expression data from patients whose class, AML (acute myeloid leukemia) or ALL (acute lymphoblastic leukemia), is known. This classifier will be used to predict the cancer class of a future patient about whom only the gene expression will be known, not the class.

The dataset consists of 72 samples. The number of gene expression levels for each patient in this dataset is 7129. In other words, 7129 attributes or features are used to describe a patient in this dataset. Since the set is relatively small, it is split into 38 training samples and 34 test samples, where each sample represents a patient [35.17]. The test set error is used for model selection here.

The classification results for the training data set, using the SG algorithm of Sect. 35.3, are summarized in Table 35.5 for $\delta = 0.01$. Results for test data are also included. Here we have seven models (A to G) for σ ranging from 32 to 20, and the design values of m vary from 12 to 38.

The CE for the training set varies from 0% to 5.26% and for the test set from 14.71% to 26.47%. Next, the behavior of the training and test error in the $(m–\sigma)$ plane is shown in Fig. 35.6. Comparing it to Fig. 35.1, we note that the training error decreases as m increases, becoming 0% at $m = n = 38$. This happens when the classification model represents exact interpolation.

Table 35.5 Classification results for the cancer gene example

Model	σ	m	Classification error (%)	
			Training	Test
A	32	12	5.26	26.47
B	30	15	2.63	20.59
C	28	21	2.63	17.65
D	26	29	0	14.71
E	24	34	0	14.71
F	22	38	0	17.65
G	20	38	0	17.65

Next, we discuss model selection based on the test CE in Table 35.5. This error first decreases with increasing m and then increases, a pattern similar to the theoretical behavior depicted in Fig. 35.1. Based on the discussion in Sect. 35.2, the best model is D. Note that here we have a somewhat degenerate case, where the training error is zero and the test error is minimum for the selected model.

35.7 Concluding Remarks

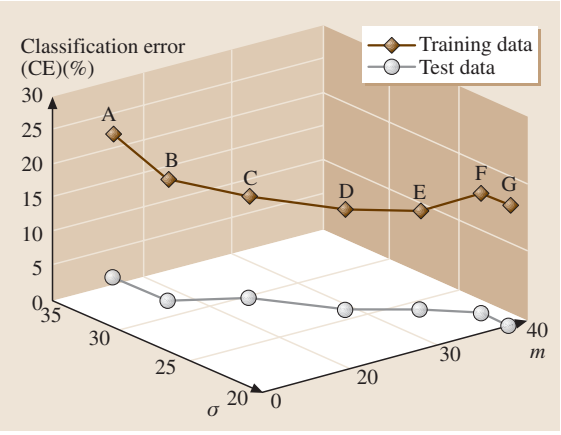


Fig. 35.6 Classification errors for the cancer gene example

In this chapter we introduced the RBF model and provided a detailed discussion of its design by evaluating the training, validation and test errors as surrogates for bias-variance phenomena. A simple example was used for illustration and then a benchmark data set was analyzed. Finally, the RBF model was used for a recent data mining application, cancer class prediction based on gene expression data. In our presentation we used only Gaussian basis functions because of their popularity and good mathematical properties. The methodology, however, is applicable to several other types of basis functions.

There is a vast body of literature on the topic of radial basis functions. The chapters in *Bishop* [35.7], *Haykin* [35.9], and *Kecman* [35.18] provide good coverage. *Buhmann* [35.19] is a rather theoretical book on this subject. A recent collection of methodologies and applications of the RBF model appears in (see *Howlett* and *Jain* [35.20]). Some other applications can be found in *Shin* and *Goel* [35.13, 21]. New developments in the theory and applications of radial basis functions can also be found in most journals and conference proceedings on neural networks and machine learning.

References

35.1 M. J. D. Powell: Radial basis functions for multivariable interpolation: A review. In: *Algorithms for Approximation*, ed. by J. C. Mason, M. G. Cox (Oxford Univ. Press, Oxford 1987) pp. 143–167

35.2 D. S. Broomhead, D. Lowe: Multivariable functional interpolation and adaptive networks, Comp. Sys. **2**, 321–355 (1988)

35.3 F. Girosi, T. Poggio: Networks and the best approximation property, Biol. Cybern. **63**, 169–176 (1990)

35.4 J. Han, M. Kamber: *Data Mining* (Morgan Kauffman, San Francisco 2001)

35.5 T. Hastie, R. Tibshirani, J. Friedman: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, Berlin Heidelberg 2001)

35.6 N. Ye: *The Handbook of Data Mining* (Lawrence Erlbaum Associates, Mahwah, NJ 2003)

35.7 C. M. Bishop: *Neural Networks for Pattern Recognition* (Oxford Univ. Press, Oxford 1995)

35.8 J. Friedman: On bias, variance, 0-1 loss, and the curse of dimensionality, Data Min. Knowl. Disc. **1**, 55–77 (1997)

35.9 S. Haykin: *Neural Networks: A Comprehensive Foundation* (Prentice Hall, New York 1999)

35.10 J. Moody, C. J. Darken: Fast learning in networks of locally-tuned processing units, Neural Comp. **1**, 281–294 (1989)

35.11 S. C. Chen, C. F. N. Cowan, P. M. Grant: Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks **2**(2), 302–309 (1991)

35.12 M. Shin: Design and Evaluation of Radial Basis Function Model for Function Approximation. Ph.D. Thesis (Syracuse Univ., Syracuse, N.Y. 1998)

35.13 M. Shin, A. L. Goel: Radial basis functions: An algebraic approach (with data mining applications), Tutorial Notes for the ECML/PKDD Conf. (ECML/PKDD, Pisa 2004)

35.14 L. Prechelt: Proben1–A Set of Neural Network Benchmark Problems and Benchmarking Rules, Interner Bericht, Universitat Karlsruhe, Fakultät für Informatik **21/94** (1994)
35.15 H. Lim: *An Empirical Study of RBF Models Using SG Algorithm* (Syracuse Univ., Syracuse, NY 2002)
35.16 S.M. Lim, K.E. Johnson (Eds.): *Methods of Microarray Data Analysis* (Kluwer, Dordrecht 2002)
35.17 T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science **286**, 531–537 (1999)
35.18 V. Kecman: *Learning and Soft Computing* (MIT Press, Cambridge 2000)
35.19 M. D. Buhmann: *Radial Basis Functions* (Cambridge Univ. Press, Cambridge 2003)
35.20 R. J. Howlett, L. C. Jain (Eds.): *Radial Basis Function Networks*, Vol. I,II (Physica, Heidelberg 2001)
35.21 M. Shin, A. L. Goel: Empirical data modeling in software engineering using radial basis functions, IEEE Trans. Software Eng. **6:26**, 567–576 (2002)

36. Data Mining Methods and Applications

In this chapter, we provide a review of the knowledge discovery process, including data handling, data mining methods and software, and current research activities. The introduction defines and provides a general background to data mining knowledge discovery in databases. In particular, the potential for data mining to improve manufacturing processes in industry is discussed. This is followed by an outline of the entire process of knowledge discovery in databases in the second part of the chapter.

The third part presents data handling issues, including databases and preparation of the data for analysis. Although these issues are generally considered uninteresting to modelers, the largest portion of the knowledge discovery process is spent handling data. It is also of great importance since the resulting models can only be as good as the data on which they are based.

The fourth part is the core of the chapter and describes popular data mining methods, separated as supervised versus unsupervised learning. In supervised learning, the training data set includes observed output values ("correct answers") for the given set of inputs. If the outputs are continuous/quantitative, then we have a regression problem. If the outputs are categorical/qualitative, then we have a classification problem. Supervised learning methods are described in the context of both regression and classification (as appropriate), beginning with the simplest case of linear models, then presenting more complex modeling with trees, neural networks, and support vector machines, and concluding with some methods, such as nearest neighbor, that are only for classification. In unsupervised learning, the training data set does not contain output values. Unsupervised learning methods are described under two categories: association rules and clustering. Association rules are appropriate for business applications where precise numerical data may not be available while clustering methods are more technically similar to the supervised learning methods presented in this chapter. Finally, this section closes with a review of various software options.

The fifth part presents current research projects, involving both industrial and business applications. In the first project, data is collected from monitoring systems, and the objective is to detect unusual activity that may require action. For example, credit card companies monitor customers' credit card usage to detect possible fraud. While methods from statistical process control were developed for similar purposes, the difference lies in the quantity of data. The second project describes data mining tools developed by Genichi Taguchi, who is well known for his industrial work on robust design. The third project tackles quality and productivity improvement in manufacturing industries. Although some detail is given, considerable research is still needed to develop a practical tool for today's complex manufacturing processes.

Finally, the last part provides a brief discussion on remaining problems and future trends.

Data mining (DM) is the process of exploration and analysis, by automatic or semiautomatic means, of large quantities of data to discover meaningful patterns and rules [36.1]. Statistical DM is exploratory data analysis with little or no human interaction using computationally feasible techniques, i. e., the attempt to find *unknown interesting* structure [36.2]. Knowledge discovery in databases (KDD) is a multidisciplinary research field for nontrivial extraction of implicit, previously unknown, and potentially useful knowledge from data [36.3]. Although some treat DM and KDD equivalently, they can be distinguished as follows. The KDD process employs DM methods (algorithms) to extract knowledge according to the specifications of measures and thresholds, using a database along with any necessary preprocessing or transformations. DM is a step in the KDD process consisting of particular algorithms (methods) that, under some acceptable objective, produces particular patterns or knowledge over the data. The two primary fields that develop DM methods are statistics and computer science. Statisticians support DM by mathematical theory and statistical methods while computer scientists develop computational algorithms and relevant software [36.4]. Prerequisites for DM include: (1) Advanced computer technology (large CPU, parallel architecture, etc.) to allow fast access to large quantities of data and enable computationally intensive algorithms and statistical methods; (2) knowledge of the business or subject matter to formulate the important business questions and interpret the discovered knowledge.

With competition increasing, DM and KDD have become critical for companies to retain customers and ensure profitable growth. Although most companies are able to collect vast amounts of business data, they are often unable to leverage this data effectively to gain new knowledge and insights. DM is the process of applying sophisticated analytical and computational techniques to discover exploitable patterns in complex data. In many cases, the process of DM results in actionable knowledge and insights. Examples of DM applications include fraud detection, risk assessment, customer relationship management, cross selling, insurance, banking, retail, etc.

While many of these applications involve customer relationship management in the service industry, a potentially fruitful area is performance improvement and cost reduction through DM in industrial and manufacturing systems. For example, in the fast-growing and highly competitive electronics industry, total revenue worldwide in 2003 was estimated to be $900 billion, and the growthrate is estimated at 8% per year (www.selectron.com). However, economies of scale, purchasing power, and global competition are making the business such that one must either be a big player or serve a niche market. Today, extremely short life cycles and constantly declining prices are pressuring the electronics industry to manufacture their products with high quality, high yield, and low production cost.

To be successful, industry will require improvements at all phases of manufacturing. Figure 36.1 illustrates the three primary phases: design, ramp-up, and production. In the production phase, maintenance of a high performance level via improved system diagnosis is needed. In the ramp-up phase, reduction in new product development time is sought by achieving the required performance as quickly as possible. Market demands have been forcing reduced development time for new product and production system design. For example, in the computer industry, a product's life cycle has been shortened to 2–3 years recently, compared to a life cycle of 3–5 years a few years ago. As a result, there are a number of new concepts in the area of production systems, such as flexible and reconfigurable manufacturing systems. Thus, in the design phase, improved system performance integrated at both the ramp-up and production phases is desired. Some of the most critical factors and barriers in the competitive development of modern manufacturing systems lie in the largely uncharted area of predicting system performance during the design phase [36.5, 6]. Consequently, current systems necessitate that a large number of design/engineering changes be made after the system has been designed.

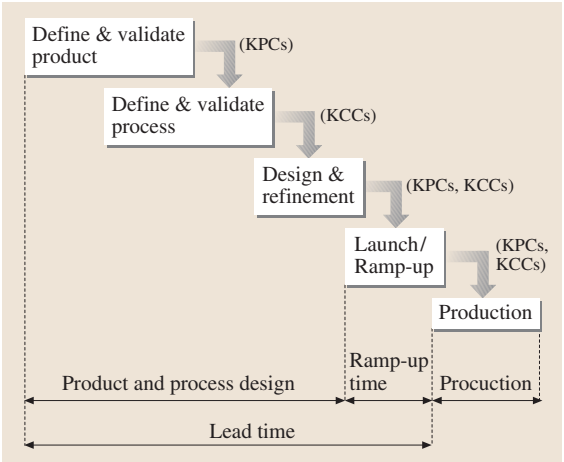


Fig. 36.1 Manufacturing system development phases. KPCs = Key product characteristics. KCCs = Key control characteristics

At all phases, system performance depends on many manufacturing process stages and hundreds or thousands of variables whose interactions are not well understood. For example, in the multi-stage printed circuit board (PCB) industry, the stages include process operations such as paste printing, chip placement, and wave soldering; and also include test operations such as optical inspection, vision inspection, and functional test. Due to advancements in information technology, sophisticated software and hardware technologies are available to record and process huge amounts of daily data in these process and testing stages. This makes it possible to extract important and useful information to improve process and product performance through DM and quality improvement technologies.

36.1 The KDD Process

The KDD process consists of four main steps:

1. Determination of business objectives,
2. Data preparation,
   a) Create target data sets,
   b) Data quality, cleaning, and preprocessing,
   c) Data reduction and projection,
3. Data mining
   a) Identify DM tasks,
   b) Apply DM tools,
4. Consolidation and application,
   a) Consolidate discovered knowledge,
   b) Implement in business decisions.

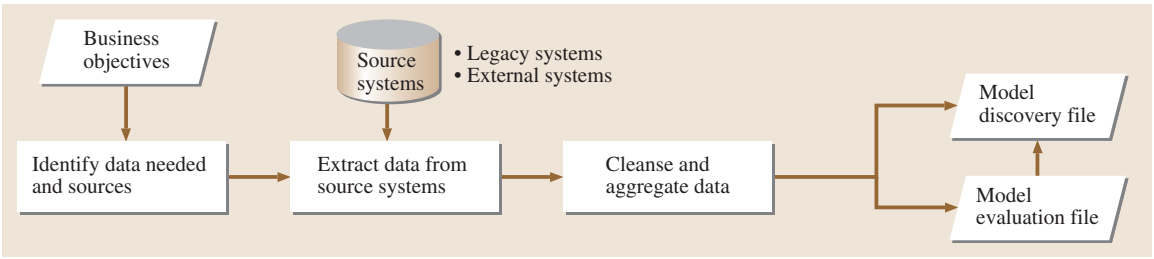


Fig. 36.2 Data preparation flow chart

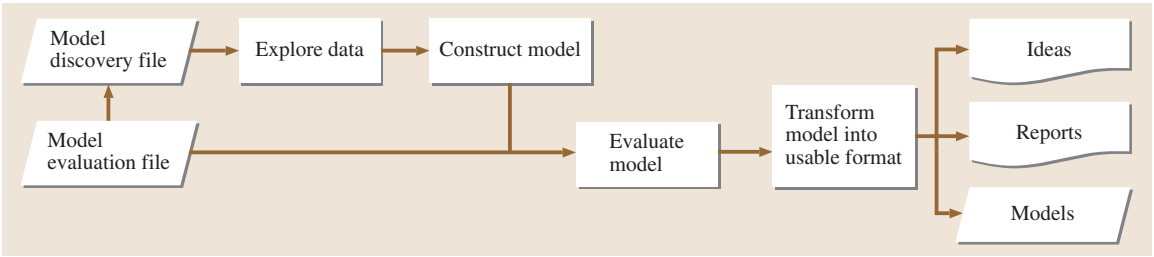


Fig. 36.3 Data mining flow chart

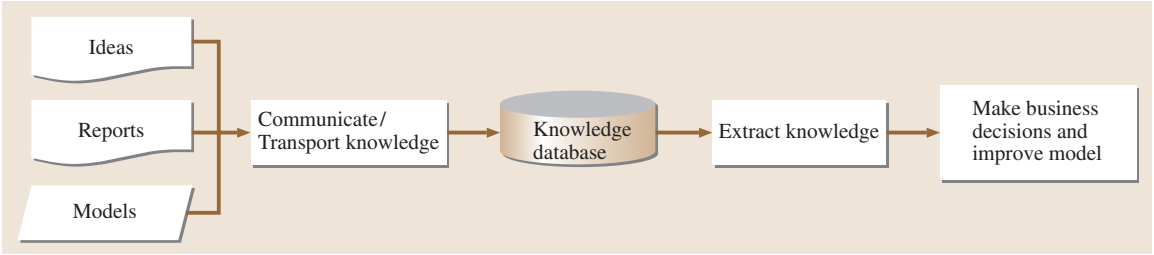


Fig. 36.4 Consolidation and application flow chart

As an example of formulating business objectives, consider a telecommunications company. It is critically important to identify those customer traits that retain profitable customers and predict fraudulent behavior, credit risks and customer churn. This knowledge may be used to improve programs in target marketing, marketing channel management, micro-marketing, and cross selling. Finally, continually updating this knowledge will enable the company to meet the challenges of new product development effectively in the future. Steps 2–4 are illustrated in figs. 36.2–36.4. Approximately 20–25% of effort is spent on determining business objectives, 50–60% of effort is spent on data preparation, 10–15% of is spent on DM, and about 10% is spent on consolidation/application.

36.2 Handling Data

The largest percentage effort of the KDD process is spent on processing and preparing the data. In this section, common forms of data storage and tools for accessing the data are described, and the important issues in data preparation are discussed.

36.2.1 Databases and Data Warehousing

A *relational database* system contains one or more objects called tables. The data or information for the database are stored in these tables. Tables are uniquely identified by their names and are comprised of columns and rows. Columns contain the column name, data type and any other attributes for the column. Rows contain the records or data for the columns. The structured query language (SQL) is the communication tool for relational database management systems. SQL statements are used to perform tasks such as updating data in a database, or retrieving data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, and Ingres. Standard SQL commands, such as *Select*, *Insert*, *Update*, *Delete*, *Create*, and *Drop*, can be used to accomplish almost everything that one needs to do with a database.

A *data warehouse* holds local databases assembled in a central facility. A *data cube* is a multidimensional array of data, where each dimension is a set of sets representing domain content, such as time or geography. The dimensions are scaled categorically, for example, region of country, state, quarter of year, week of quarter. The cells of the cube contain aggregated measures (usually counts) of variables. To explore the data cube, one can *drill down*, *drill up*, and *drill through*. Drill down involves splitting an aggregation into subsets, e.g., splitting region of country into states. Drill up involves consolidation, i. e., aggregating subsets along a dimension. Drill through involves subsets crossing multiple sets, e.g., the user might investigate statistics within a state subset by time. Other databases and tools include object-oriented databases, transactional databases, time series and spatial databases, online analytical processing (OLAP), multidimensional OLAP (MOLAP), and relational OLAP using extended SQL (ROLAP). See Chapt. 2 of *Han* and *Kamber* [36.7] for more details.

36.2.2 Data Preparation

The purpose of this step in the KDD process is to identify data quality problems, sources of noise, data redundancy, missing data, and outliers. Data quality problems can involve inconsistency with external data sets, uneven quality (e.g., if a respondent fakes an answer), and biased opportunistically collected data. Possible sources of noise include faulty data collection instruments (e.g., sensors), transmission errors (e.g., intermittent errors from satellite or internet transmissions), data entry errors, technology limitations errors, misused naming conventions (e.g., using the same names for different meanings), and incorrect classification.

Redundant data exists when the same variables have different names in different databases, when a raw variable in one database is a derived variable in another, and when changes in a variable over time are not reflected in the database. These irrelevant variables impede the speed of the KDD process because dimension reduction is needed to eliminate them. Missing data may be irrelevant if we can extract useful knowledge without imputing the missing data. In addition, most statistical methods for handling missing data may fail for massive data sets, so new or modified methods still need to be developed. In detecting outliers, sophisticated methods like the Fisher information matrix or convex hull peeling are available, but are too complex for massive data sets. Although outliers may be easy to visualize in low dimensions, high-dimensional outliers may not show up in low-dimensional projections.

Currently, clustering and other statistical modeling are used.

The data preparation process involves three steps: data cleaning, database sampling, and database reduction and transformation. Data cleaning includes removal of duplicate variables, imputation of missing values, identification and correction of data inconsistencies, identification and updating of stale data, and creating a unique record (case) identification (ID). Via database sampling, the KDD process selects appropriate parts of the databases to be examined. For this to work, the data must satisfy certain conditions (e.g., no systematic biases). The sampling process can be expensive if the data have been stored in a database system such that it is difficult to sample the data the way you want and many operations need to be executed to obtain the targeted data. One must balance a trade-off between the costs of the sampling process and the mining process. Finally, database reduction is used for data cube aggregation, dimension reduction, elimination of irrelevant and redundant attributes, data compression, and encoding mechanisms via quantizations, wavelet transformation, principle components, etc.

36.3 Data Mining (DM) Models and Algorithms

The DM process is illustrated in Fig. 36.5. In this process, one will start by choosing an appropriate class of models. To fit the best model, one needs to split the sample data into two parts: the training data and the testing data. The training data will be used to fit the model and the testing data is used to refine and tune the fitted model. After the final model is obtained, it is recommended to use an independent data set to evaluate the goodness of the final model, such as comparing the prediction error to the accuracy requirement. (If independent data are not available, one can use the cross-validation method to compute prediction error.) If the accuracy requirement is not satisfied, then one must revisit earlier steps to reconsider other classes of models or collect additional data.

Before implementing any sophisticated DM methods, data description and visualization are used for initial exploration. Tools include descriptive statistical measures for central tendency/location, dispersion/spread, and distributional shape and symmetry; class characterizations and comparisons using analytical approaches, attribute relevance analysis, and class discrimination and comparisons; and data visualization using scatter-plot matrices, density plots, 3-D stereoscopic scatter-plots, and parallel coordinate plots. Following this initial step, DM methods take two forms: supervised versus unsupervised learning. Supervised learning is described as
textitlearning with a teacher, where the *teacher* provides data with correct answers. For example, if we want to classify online shoppers as buyers or non-buyers using an available set of variables, our data would include actual instances of buyers and non-buyers for training a DM method. Unsupervised learning is described as

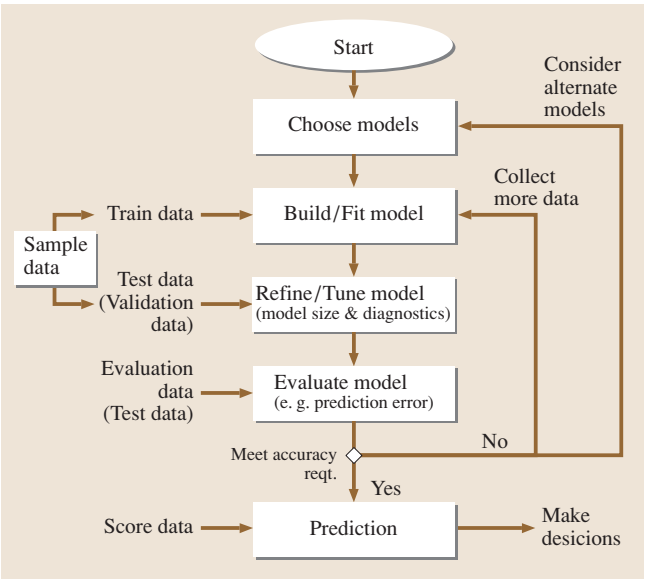


Fig. 36.5 Data mining process

learning without a teacher. In this case, correct answers are not available, and DM methods would search for patterns or clusters of similarity that could later be linked to some explanation.

36.3.1 Supervised Learning

In supervised learning, we have a set of *input* variables (also known as predictors, independent variables, $\boldsymbol{x}$) that are measured or preset, and a set of *output* variables (also known as responses, dependent variables, $\boldsymbol{y}$) that are measured and assumed to be influenced by the in-

puts. If the outputs are continuous/quantitative, then we have a *regression* or *prediction* problem. If the outputs are categorical/qualitative, then we have a *classification* problem. First, a DM model/system is established based on the collected input and output data. Then, the established model is used to predict output values at new input values. The predicted values are denoted by $\hat{\boldsymbol{y}}$.

The DM perspective of *learning with a teacher*, follows these steps:

- Student presents an answer ($\hat{\boldsymbol{y}}_i$ given $\boldsymbol{x}_i$);
- Teacher provides the correct answer $\boldsymbol{y}_i$ or an error $\boldsymbol{e}_i$ for the student's answer;
- The result is characterized by some *loss function* or *lack-of-fit criterion*:LOF($\boldsymbol{y}$, $\hat{\boldsymbol{y}}$);
- The objective is to minimize the expected loss.

Supervised learning includes the common engineering task of function approximation, in which we assume that the output is related to the input via some function $f(\boldsymbol{x}, \epsilon)$, where ϵ represents a random error, and seek to approximate $f(\cdot)$.

Below, we describe several supervised learning methods. All can be applied to both the regression and classification cases, except for those presented below under Other Classifikation Methods. We maintain the following notation. The j-th input variable is denoted by x_j (or random variable X_j) and the corresponding boldface $\boldsymbol{x}$ (or $\boldsymbol{X}$) denotes the vector of p input variables $(x_1, x_2, \ldots, x_p)^{\mathrm{T}}$, where boldface $\boldsymbol{x}_i$ denotes the i-th sample point; N is the number of sample points, which corresponds to the number of observations of the response variable; the response variable is denoted by y (or random variable Y), where y_i denotes the i-th response observation. For the regression case, the response y is quantitative, while for the classification case, the response values are indices for C classes ($c = 1, \ldots, C$). An excellent reference for these methods is *Hastie* et al. [36.8].

Linear and Additive Methods

In the regression case, the basic linear method is simply the *multiple linear* regression model form

$$\mu(\boldsymbol{x}; \beta) = E[Y \mid \boldsymbol{X} = \boldsymbol{x}] = \beta_0 + \sum_{m=1}^{M} \beta_m b_m(\boldsymbol{x}),$$

where the model terms $b_m(\boldsymbol{x})$ are pre-specified functions of the input variables, for example, a simple linear term $b_m(\boldsymbol{x}) = x_j$ or a more complex interaction term $b_m(\boldsymbol{x}) = x_j x_k^2$. The key is that the model is *linear in the parameters* $\boldsymbol{\beta}$. Textbooks that cover linear regression are abundant (e.g., [36.9, 10]). In particular, *Neter* et al. [36.11] provides a good background on residual analysis, model diagnostics, and model selection using best subsets and stepwise methods. In model selection, insignificant model terms are eliminated; thus, the final model may be a subset of the original pre-specified model. An alternate approach is to use a shrinkage method that employs a penalty function to *shrink* estimated model parameters towards zero, essentially reducing the influence of less important terms. Two options are ridge regression [36.12], which uses the penalty form $\sum \beta_m^2$, and the lasso [36.13], which uses the penalty form $\sum |\beta_m|$.

In the classification case, linear methods generate *linear decision boundaries* to separate the C classes. Although a direct linear regression approach could be applied, it is known not to work well. A better method is *logistic regression* [36.14], which uses log-odds (or logit transformations) of the posterior probabilities $\mu_c(\boldsymbol{x}) = P(Y = c|\boldsymbol{X} = \boldsymbol{x})$ for classes $c = 1, \ldots, C-1$ in the form

$$\log \frac{\mu_c(\boldsymbol{x})}{\mu_C(\boldsymbol{x})} = \log \frac{P(Y = c|\boldsymbol{X} = \boldsymbol{x})}{P(Y = C|\boldsymbol{X} = \boldsymbol{x})} = \beta_{c0} + \sum_{j=1}^{p} \beta_{cj} x_j \,,$$

where the C posterior probabilities $\mu_c(\boldsymbol{x})$ must sum to one. The decision boundary between class $c < C$ and class C is defined by the hyperplane $\{\boldsymbol{x} \mid \beta_{c0} + \sum \beta_{cj} x_j = 0\}$, where the log-odds are zero. Similarly, the decision boundary between classes $c \neq C$ and $d \neq C$, derived from the log-odds for classes c and d, is defined by $\{\boldsymbol{x} \mid \beta_{c0} + \sum \beta_{cj} x_j = \beta_{d0} + \sum \beta_{dj} x_j\}$. In the binary case ($C = 2$), if we define $\mu(\boldsymbol{x}) = P(Y = 1|\boldsymbol{X} = \boldsymbol{x})$, then $1 - \mu(\boldsymbol{x}) = P(Y = 2|\boldsymbol{X} = \boldsymbol{x})$. The logit transformation is then defined as $g(\mu) = \mu/(1 - \mu)$.

Closely related to logistic regression is *linear discriminant analysis* [36.15], which utilizes exactly the same linear form for the log-odds ratio, and defines linear *discriminant functions* $\delta_c(\boldsymbol{x})$, such that $\boldsymbol{x}$ is classified to class c if its maximum discriminant is $\delta_c(\boldsymbol{x})$. The difference between the two methods is how the parameters are estimated. Logistic regression maximizes the conditional likelihood involving the posterior probabilities $P(Y = c|\boldsymbol{X})$ while linear discriminant analysis maximizes the full log-likelihood involving the unconditional probabilities $P(Y = c, \boldsymbol{X})$. More general forms of discriminant analysis are discussed below under Other Classifikation Methods.

Finally, it should be noted that the logistic regression model is one form of *generalized linear model*

(GLM) [36.16]. GLM forms convert what appear to be nonlinear models into linear models, using tools such as transformations (e.g., logit) or conditioning on nonlinear parameters. This then enables the modeler to use traditional linear modeling analysis techniques. However, real data often do not satisfy the restrictive conditions of these models.

Rather than using pre-specified model terms, as in a linear model, a *generalized additive model* (GAM) [36.17], provides a more flexible statistical method to enable modeling of nonlinear patterns in each input dimension. In the regression case, the basic GAM form is

$$\mu(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{p} f_j(x_j) \,,$$

where the $f_j(\cdot)$ are unspecified (smooth) univariate functions, one for each input variable. The additive restriction prohibits inclusion of any interaction terms. Each function is fitted using a *nonparametric regression* modeling method, such as running-line smoothers (e.g., *lowess*, [36.18]), smoothing splines or kernel smoothers [36.19–21]. In the classification case, an additive logistic regression model utilizes the logit transformation for classes $c = 1, \ldots, C-1$ as above

$$\begin{aligned} \log \frac{\mu_c(\boldsymbol{x})}{\mu_C(\boldsymbol{x})} &= \log \frac{P(Y=c|\boldsymbol{X}=\boldsymbol{x})}{P(Y=C|\boldsymbol{X}=\boldsymbol{x})} \\ &= \beta_0 + \sum_{j=1}^{p} f_j(x_j) \,, \end{aligned}$$

where an additive model is used in place of the linear model. However, even with the flexibility of nonparametric regression, GAM may still be too restrictive. The following sections describe methods that have essentially no assumptions on the underlying model form.

Trees and Related Methods

One DM decision tree model is *chi-square automatic interaction detection* (CHAID) [36.22, 23], which builds non-binary trees using a chi-square test for the classification case and an F-test for the regression case. The CHAID algorithm first creates categorical input variables out of any continuous inputs by dividing them into several categories with approximately the same number of observations. Next, input variable categories that are not statistically different are combined, while a Bonferroni p-value is calculated for those that are statistically different. The best split is determined by the smallest p-value. CHAID continues to select splits until the smallest p-value is greater than a pre-specified significance level (α).

The popular *classification and regression trees* (CART) [36.24] utilize recursive partitioning (binary splits), which evolved from the work of *Morgan* and *Sonquist* [36.25] and *Fielding* [36.26] on analyzing survey data. CARTs have a forward stepwise procedure that adds model terms and backward procedure for pruning. The model terms partition the $\boldsymbol{x}$-space into disjoint hyper-rectangular regions via indicator functions: $b^+(x;t) = 1\{x > t\}$, $b^-(x;t) = 1\{x \le t\}$, where the *split-point* t defines the borders between regions. The resulting model terms are:

$$f_m(\boldsymbol{x}) = \prod_{l=1}^{L_m} b^{s_{l,m}}(x_{v(l,m)}; t_{l,m}) \,, \tag{36.1}$$

where, L_m is the number of univariate indicator functions multiplied in the m-th model term, $x_{v(l,m)}$ is the input variable corresponding to the l-th indicator function in the m-th model term, $t_{l,m}$ is the split-point corresponding to $x_{v(l,m)}$, and $s_{l,m}$ is $+1$ or -1 to indicate the direction of the partition. The CART model form is then

$$f(\boldsymbol{x};\beta) = \beta_0 + \sum_{m=1}^{M} \beta_m f_m(\boldsymbol{x}) \,. \tag{36.2}$$

The partitioning of the $\boldsymbol{x}$-space does not keep the parent model terms because they are redundant. For example, suppose the current set has the model term:

$$f_a(\boldsymbol{x}) = 1\{x_3 > 7\} \cdot 1\{x_4 \le 10\} \,,$$

and the forward stepwise algorithm chooses to add

$$\begin{aligned} f_b(\boldsymbol{x}) &= f_a(\boldsymbol{x}) \cdot 1\{x_5 > 13\} \\ &= 1\{x_3 > 7\} \cdot 1\{x_4 \le 10\} \cdot 1\{x_5 > 13\} \,. \end{aligned}$$

Then the model term $f_a(\boldsymbol{x})$ is dropped from the current set. Thus, the recursive partitioning algorithm follows a binary tree with the current set of model terms $f_m(\boldsymbol{x})$ consisting of the M leaves of the tree, each of which corresponds to a different region R_m.

In the regression case, CART minimizes the squared error loss function,

$$\mathrm{LOF}(\hat{f}) = \sum_{i=1}^{N} \left[y_i - \hat{f}(\boldsymbol{x}_i)\right]^2 \,,$$

and the approximation is a piecewise-constant function. In the classification case, each region R_m is classified into one of the C classes. Specifically, define the

proportion of class c observations in region R_m as

$$\hat{\delta}_{mc} = \frac{1}{N_m} \sum_{\boldsymbol{x}_i \in R_m} 1\{y_i = c\} ,$$

where N_m is the number of observations in the region R_m. Then the observations in region R_m are classified into the class c corresponding to the maximum proportion $\hat{\delta}_{mc}$. The algorithm is exactly the same as for regression, but with a different loss function. Appropriate choices include minimizing the misclassification error (i. e., the number of misclassified observations), the Gini index, $\sum_{c=1}^{C} \hat{\delta}_{mc}(1-\hat{\delta}_{mc})$, or the deviance $\sum_{c=1}^{C} \hat{\delta}_{mc}\log(\hat{\delta}_{mc})$.

The exhaustive search algorithms for CART simultaneously conduct variable selection (x) and split-point selection (t). To reduce computational effort, the *fast algorithm for classification trees* [36.27] separates the two tasks. At each existing model term (leaf of the tree), F-statistics are calculated for variable selection. Then linear discriminant analysis is used to identify the split-point. A version for logistic and Poisson regression was presented by *Chaudhuri* et al. [36.28].

The primary drawback of CART and FACT is a bias towards selecting higher-order interaction terms due to the property of keeping only the leaves of the tree. As a consequence, these tree methods do not provide robust approximations and can have poor prediction accuracy. *Loh* and *Shih* [36.29] address this issue for FACT with a variant of their classification algorithm called QUEST that clusters classes into superclasses before applying linear discriminant analysis. For CART, *Friedman* et al. [36.30] introduced to the statistics literature the concepts of *boosting* [36.31] and *bagging* [36.32] from the machine learning literature. The bagging approach generates many bootstrap samples, fits a tree to each, then uses their average prediction. In the framework of boosting, a model term, called a *base learner*, is a *small* tree with only L disjoint regions (L is selected by the user), call it $B(\boldsymbol{x}, \boldsymbol{a})$, where $\boldsymbol{a}$ is the vector of tree coefficients. The boosting algorithm begins by fitting a small tree $B(\boldsymbol{x}, \boldsymbol{a})$ to the data, and the first approximation, $\hat{f}_1(\boldsymbol{x})$, is then this first small tree. In the m-th iteration, residuals are calculated, then a small tree $B(\boldsymbol{x}, \boldsymbol{a})$ is fitted to the residuals and combined with the latest approximation to create the m-th approximation:

$$\hat{f}_m(\boldsymbol{x}; \beta_0, \beta_1, \ldots, \beta_m) = \hat{f}_{m-1}(\boldsymbol{x}; \beta_0, \beta_1, \ldots, \beta_{m-1}) + \beta_m B(\boldsymbol{x}, \boldsymbol{a}) ,$$

where a line search is used to solve for β_m. The resulting boosted tree, called a *multiple additive regression tree* (MART) [36.33], then consists of much lower-order interaction terms. *Friedman* [36.34] presents *stochastic gradient boosting*, with a variety of loss functions, in which a bootstrap-like bagging procedure is included in the boosting algorithm.

Finally, for the regression case only, *multivariate adaptive regression splines* (MARS) [36.35] evolved from CART as an alternative to its piecewise-constant approximation. Like CART, MARS utilizes a forward stepwise algorithm to select model terms followed by a backward procedure to prune the model. A univariate version (appropriate for additive relationships) was presented by *Friedman* and *Silverman* [36.36]. The MARS approximation bends to model curvature at *knot* locations, and one of the objectives of the forward stepwise algorithm is to select appropriate knots. An important difference from CART is that MARS maintains the parent model terms, which are no longer redundant, but are simply lower-order terms.

MARS model terms have the same form as (36.1), except the indicator functions are replaced with truncated linear functions,

$$[b^+(x; t) = [+(x-t)]_+, \quad b^-(x; t) = [-(x-t)]_+ ,$$

where $[q]_+ = \max(0, q)$ and t is an univariate knot. The search for new model terms can be restricted to interactions of a maximum order (e.g., $L_m \leq 2$ permits up through two-factor interactions). The resulting MARS approximation, following (36.2), is a continuous, piecewise-linear function. After selection of the model terms is completed, smoothness to achieve a certain degree of continuity may be applied.

Hastie et al. [36.8] demonstrate significant improvements in accuracy using MART over CART. For the regression case, comparisons between MART and MARS yield comparable results [36.34]. Thus, the primary decision between these two methods is whether a piecewise-constant approximation is satisfactory or if a continuous, smooth approximation would be preferred.

Artificial Neural Networks

Artificial neural network (ANN) models have been very popular for modeling a variety of physical relationships (for a general introduction see *Lippmann* [36.37] or *Haykin* [36.38]; for statistical perspectives see *White* [36.39], *Baron* et al. [36.40], *Ripley* [36.23], or *Cheng* and *Titterington* [36.41]). The original motivation for ANNs comes from how *learning* strengthens connections along neurons in the brain. Commonly, an ANN model is represented by a diagram of nodes in vari-

ous layers with weighted connections between nodes in different layers (Fig. 36.6). At the input layer, the nodes are the input variables and at the output layer, the nodes are the response variable(s). In between, there is usually at least one *hidden* layer which induces flexibility into the modeling. *Activation functions* define transformations between layers (e.g., input to hidden). Connections between nodes can *feed back* to previous layers, but for supervised learning, the typical ANN is *feedforward* only with at least one hidden layer.

The general form of a feedforward ANN with one hidden layer and activation functions $b_1(\cdot)$ (input to hidden) and $b_2(\cdot)$ (hidden to output) is

$$f_c(\boldsymbol{x};\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\gamma_c) = b_2\left[\sum_{h=1}^{H} w_{hc}\cdot b_1\left(\sum_{j=1}^{p} v_{jh}x_j+\theta_h\right)+\gamma_c\right], \tag{36.3}$$

where $c = 1,\ldots,C$ and C is the number of output variables, p is the number of input variables, H is the number of hidden nodes, the weights v_{jh} link input nodes j to hidden nodes h and w_{hc} link hidden nodes h to output nodes c, and θ_h and γ_c are constant terms called bias nodes (like intercept terms). The number of coefficients to be estimated is $(p+1)H+(H+1)C$, which is often larger than N. The simplest activation function is a linear function $b(z) = z$, which reduces the ANN model in (36.3) with one response variable to a multiple linear regression equation. For more flexibility, the recommended activation functions between the input and hidden layer(s) are the S-shaped *sigmoidal* functions or

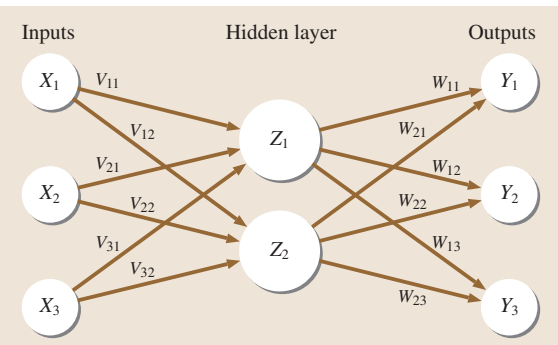


Fig. 36.6 Diagram of a typical artificial neural network for function approximation. The input nodes correspond to the input variables, and the output node(s) correspond to the output variable(s). The number of hidden nodes in the hidden layer must be specified by the user

the bell-shaped *radial basis functions*. Commonly used sigmoidal functions are the logistic function

$$b(z) = \frac{1}{1+\mathrm{e}^{-z}}$$

and the hyperbolic tangent

$$b(z) = \tanh(z) = \frac{1-\mathrm{e}^{-2x}}{1+\mathrm{e}^{-2x}}\,.$$

The most common radial basis function is the Gaussian probability density function.

In the regression case, each node in the output layer represents a quantitative response variable. The output activation function may be either a linear, sigmoidal, or radial basis function. Using a logistic activation function from input to hidden and from hidden to output, the ANN model in (36.3) becomes

$$f_c(\boldsymbol{x};\boldsymbol{w},\boldsymbol{v},\boldsymbol{\theta},\gamma_c) = \left[1+\exp\left(-\sum_{h=1}^{H} w_{hc}z_h+\gamma_c\right)\right]^{-1},$$

where for each hidden node h

$$z_h = \left[1+\exp\left(-\sum_{j=1}^{p} v_{jh}x_j+\theta_h\right)\right]^{-1}.$$

In the classification case with C classes, each class is represented by a different node in the output layer. The recommended output activation function is the *softmax* function. For output node c, this is defined as

$$b(z_1,\ldots,z_C;c) = \frac{\mathrm{e}^{z_c}}{\sum_{d=1}^{C}\mathrm{e}^{z_d}}\,.$$

This produces output values between zero and one that sum to one and, consequently, permits the output values to be interpreted as posterior probabilities for a categorical response variable.

Mathematically, an ANN model is a nonlinear statistical model, and a nonlinear method must be used to estimate the coefficients (weights v_{jh} and w_{hc}, biases θ_h and γ_c) of the model. This estimation process is called network training. Typically, the objective is to minimize the squared error lack-of-fit criterion

$$\mathrm{LOF}(\hat{f}) = \sum_{c=1}^{C}\sum_{i=1}^{N}\left[y_i-\hat{f}_c(\boldsymbol{x}_i)\right]^2.$$

The most common method for training is *backpropagation*, which is based on gradient descent. At each

iteration, each coefficient (say w) is adjusted according to its contribution to the lack-of-fit

$$\Delta w = \alpha \frac{\partial(\text{LOF})}{\partial w} ,$$

where the user-specified α controls the step size; see *Rumelhart* et al. [36.42] for more details. More efficient training procedures are a subject of current ANN research.

Another major issue is the network *architecture*, defined by the number of hidden nodes. If too many hidden nodes are permitted, the ANN model will overfit the data. Many model discrimination methods have been tested, but the most reliable is validation of the model on a testing data set separate from the training data set. Several ANN architectures are fitted to the training data set and then prediction error is measured on the testing data set. Although ANNs are generally flexible enough to model anything, they are computationally intensive, and a significant quantity of representative data is required to both fit and validate the model. From a statistical perspective, the primary drawback is the overly large set of coefficients, none of which provide any intuitive understanding for the underlying model structure. In addition, since the nonlinear model form is not motivated by the true model structure, too few training data points can result in ANN approximations with extraneous nonlinearity. However, given enough good data, ANNs can outperform other modeling methods.

Support Vector Machines

Referring to the linear methods for classification described earlier, the decision boundary between two classes is a hyperplane of the form $\{\boldsymbol{x} \mid \beta_0 + \sum \beta_j x_j = 0\}$. The *support vectors* are the points that are most critical to determining the optimal decision boundary because they lie close to the points belonging to the other class. With *support vector machines* (SVM) [36.43,44], the linear decision boundary is generalized to the more flexible form

$$f(\boldsymbol{x};\boldsymbol{\beta}) = \beta_0 + \sum_{m=1}^{M} \beta_m g_m(\boldsymbol{x}), \qquad (36.4)$$

where the $g_m(\boldsymbol{x})$ are transformations of the input vector. The decision boundary is then defined by $\{\boldsymbol{x} \mid f(\boldsymbol{x};\boldsymbol{\beta}) = 0\}$. To solve for the optimal decision boundary, it turns out that we do not need to specify the transformations $g_m(\boldsymbol{x})$, but instead require only the *kernel function* [36.21,45]:

$$K(\boldsymbol{x},\boldsymbol{x}') = \langle [g_1(\boldsymbol{x}), \ldots, g_M(\boldsymbol{x})], [g_1(\boldsymbol{x}'), \ldots, g_M(\boldsymbol{x}')] \rangle .$$

Two popular kernel functions for SVM are polynomials of degree d, $K(\boldsymbol{x},\boldsymbol{x}') = (1 + \langle \boldsymbol{x},\boldsymbol{x}' \rangle)^d$, and radial basis functions, $K(\boldsymbol{x},\boldsymbol{x}\prime) = \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/c)$.

Given $K(\boldsymbol{x},\boldsymbol{x}')$, we maximize the following Lagrangian dual-objective function:

$$\max_{\alpha_1,\ldots\alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} K(\boldsymbol{x}_i, \boldsymbol{x}_i')$$
$$\text{s.t.} \quad 0 \le \alpha_i \le \gamma , \text{ for } i = 1, \ldots, N \text{ and}$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0 ,$$

where γ is an SVM tuning parameter. The optimal solution allows us to rewrite $f(\boldsymbol{x};\boldsymbol{\beta})$ as

$$f(\boldsymbol{x};\boldsymbol{\beta}) = \beta_0 + \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x},\boldsymbol{x}_i) ,$$

where β_0 and $\alpha_1, \ldots, \alpha_N$ are determined by solving $f(\boldsymbol{x};\boldsymbol{\beta}) = 0$. The support vectors are those $\boldsymbol{x}_i$ corresponding to nonzero α_i. A smaller SVM tuning parameter γ leads to more support vectors and a smoother decision boundary. A testing data set may be used to determine the best value for γ.

The SVM extension to more than two classes solves multiple two-class problems. SVM for regression utilizes the model form in (36.4) and requires specification of a loss function appropriate for a quantitative response [36.8,46]. Two possibilities are the ϵ-insensitive function

$$V_\epsilon(e) = \begin{cases} 0 & \text{if } |e| < \epsilon , \\ |e| - \epsilon & \text{otherwise} , \end{cases}$$

which ignores errors smaller than ϵ, and the *Huber* [36.47] function

$$V_{\text{H}}(e) = \begin{cases} e^2/2 & \text{if } |e| \le 1.345 , \\ 1.345|e| - e^2/2 & \text{otherwise} , \end{cases}$$

which is used in robust regression to reduce model sensitivity to outliers.

Other Classification Methods

In this section, we briefly discuss some other concepts that are applicable to DM classification problems. The basic intuition behind a good classification method is derived from the *Bayes classifier*, which utilizes the posterior distribution $P(Y = c|\boldsymbol{X} = \boldsymbol{x})$. Specifically, if $P(Y = c|\boldsymbol{X} = \boldsymbol{x})$ is the maximum over $c = 1, \ldots, C$, then $\boldsymbol{x}$ would be classified to class c.

Nearest neighbor (NN) [36.48] classifiers seek to estimate the Bayes classifier directly without specification of any model form. The k-NN classifier identifies the k closest points to $\boldsymbol{x}$ (using Euclidean distance) as the *neighborhood* about $\boldsymbol{x}$, then estimates $P(Y = c|\boldsymbol{X} = \boldsymbol{x})$ with the fraction of these k points that are of class c. As k increases, the decision boundaries become smoother; however, the *neighborhood* becomes less local (and less relevant) to $\boldsymbol{x}$. This problem of local representation is even worse in high dimensions, and modifications to the distance measure are needed to create a practical k-NN method for DM. For this purpose, *Hastie* and *Tibshirani* [36.49] proposed the discriminant adaptive NN distance measure to reshape the neighborhood adaptively at a given $\boldsymbol{x}$ to capture the critical points to distinguish between the classes.

As mentioned earlier, linear discriminant analysis may be too restrictive in practice. *Flexible discriminant analysis* replaces the linear decision boundaries with more flexible regression models, such as GAM or MARS. *Mixture discriminant analysis* relaxes the assumption that that classes are more or less spherical in shape by allowing a class to be represented by multiple (spherical) clusters; see *Hastie* et al. [36.50] and *Ripley* [36.23] for more details.

K-means clustering classification applies the K-means clustering algorithm separately to the data for each of the C classes. Each class c will then be represented by K clusters of points. Consequently, nonspherical classes may be modeled. For a new input vector $\boldsymbol{x}$, determine the closest cluster, then assign $\boldsymbol{x}$ to the the class associated with that cluster.

Genetic algorithms [36.51, 52] use processes such as genetic combination, mutation, and natural selection in an optimization based on the concepts of natural evolution. One generation of models competes to pass on characteristics to the next generation of models, until the best model is found. Genetic algorithms are useful in guiding DM algorithms, such as neural networks and decision trees [36.53].

36.3.2 Unsupervised Learning

In unsupervised learning, correct answers are not available, so there is no clear measure of success. Success must be judged subjectively by the value of discovered knowledge or the effectiveness of the algorithm. The statistical perspective is to observe N vectors from the population distribution, then conduct direct inferences on the properties (e.g. relationship, grouping) of the population distribution. The number of variables or attributes is often very high (much higher than that in supervised learning). In describing the methods, we denote the j-th variable by x_j (or random variable X_j), and the corresponding boldface $\boldsymbol{x}$ (or $\boldsymbol{X}$) denotes the vector of p variables $(x_1, x_2, \ldots, x_p)^{\mathrm{T}}$, where boldface $\boldsymbol{x}_i$ denotes the i-th sample point. These variables may be either quantitative or qualitative.

Association Rules

Association rules or *affinity groupings* seek to find associations between the values of the variables $\boldsymbol{X}$ that provide knowledge about the population distribution. *Market basket analysis* is a well-known special case, for which the extracted knowledge may be used to link specific products. For example, consider all the items that may be purchased at a store. If the analysis identifies that items A and B are commonly purchased together, then sales promotions could exploit this to increase revenue.

In seeking these associations, a primary objective is to identify variable values that occur together with high probability. Let S_j be the the set of values for X_j, and consider a subset $s_j \subseteq S_j$. Then we seek subsets $s_1, \ldots, s_p$ such that

$$P\left[\bigcap_{j=1}^{p}(X_j \in s_j)\right] \tag{36.5}$$

is large. In market basket analysis, the variables $\boldsymbol{X}$ are converted to a set of binary variables $\boldsymbol{Z}$, where each attainable value of each X_j corresponds to a variable Z_k. Thus, the number of Z_k variables is $K = \sum |S_j|$. If binary variable Z_k corresponds to $X_j = v$, then $Z_k = 1$ when $X_j = v$ and $Z_k = 0$ otherwise. An *item set* κ is a realization of $\boldsymbol{Z}$. For example, if the Z_k represent the possible products that could be purchased from a store, then an item set would be the set of items purchased together by a customer. Note that the number of $Z_k = 1$ in an item set is at most p. Equation (36.5) now becomes

$$P\left[\bigcap_{k\in\kappa}(Z_k = 1)\right],$$

which is estimated by

$$T(\kappa) = \frac{\text{Number of observations for which item set } \kappa \text{ occurs}}{N}.$$

$T(\kappa)$ is called the *support* for the rule. We can select a lower bound t such that item sets with $T(\kappa) > t$ would be considered to have large support.

Further knowledge may be extracted via the a priori algorithm [36.54] in the form of if–then statements. For an item set κ, the items with $Z_k = 1$ would be partitioned into two disjoint item subsets such that $A \bigcup B = \kappa$. The association rule would be stated as "if A, then B" and denoted by $A \Rightarrow B$, where A is called the *antecedent* and B is called the *consequent*. This rule's support $T(A \Rightarrow B)$ is the same as $T(\kappa)$ calculated above, an estimate of the joint probability. The *confidence* or *predictability* of this rule is

$$C(A \Rightarrow B) = \frac{T(A \Rightarrow B)}{T(A)} ,$$

which is an estimate of the conditional probability $P(B|A)$. The *expected confidence* is the support of B, $T(B)$, and an estimate for the unconditional probability $P(B)$. The *lift* is the ratio of the confidence over the expected confidence,

$$L(A \Rightarrow B) = \frac{C(A \Rightarrow B)}{T(B)} ,$$

which, if greater than , can be interpreted as the increased prevalence of B when associated with A. For example, if $T(B) = 5\%$, then B is estimated to occur unconditionally 5% of the time. If $C(A \Rightarrow B) = 40\%$, then given A occurs, B is estimated to occur 40% of the time. This results in a lift of 8, implying that B is 8 times more likely to occur if we know that A occurs.

Cluster Analysis

The objective of cluster analysis is to partition the N observations of $\boldsymbol{x}$ into groups or clusters such that the dissimilarities within each cluster are smaller than the dissimilarities between different clusters [36.55]. Typically the variables $\boldsymbol{x}$ are all quantitative, and a distance measure (e.g., Euclidean) is used to measure dissimilarity. For categorical x variables, a dissimilarity measure must be explicitly defined. Below, we describe some of the more common methods.

K-means [36.56] is the best-known clustering tool. It is appropriate when the variables $\boldsymbol{x}$ are quantitative. Given a prespecified value K, the method partitions the N observations of $\boldsymbol{x}$ into exactly K clusters by minimizing within-cluster dissimilarity. Squared Euclidean distance

$$d(\boldsymbol{x}_i, \boldsymbol{x}_i') = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

is used to measure dissimilarity. For a specific clustering assignment $\boldsymbol{C} = (C_1, \ldots, C_K)$, the within-cluster dissimilarity is measured by calculating $d(\boldsymbol{x}_i, \boldsymbol{x}_i')$ for all points $\boldsymbol{x}_i, \boldsymbol{x}_i'$ within a cluster C_k, then summing over the K clusters. This is equivalent to calculating

$$W(\boldsymbol{C}) = \sum_{k=1}^{K} \sum_{i \in C_k} d(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_k) ,$$

where the cluster mean $\bar{\boldsymbol{x}}_k$ is the sample mean vector of the points in cluster C_k. Given a current set of cluster means, the K-means algorithm assigns each point to the closest cluster mean, calculates the new cluster means, and iterates until the cluster assignments do not change.

Unfortunately, because of its dependence on the squared Euclidean distance measure, K-means clustering is sensitive to outliers (i. e., is not robust). *K-mediods* [36.57] is a generalized version that utilizes an alternately defined cluster *center* in place of the cluster means and an alternate distance measure.

Density-based clustering (DBSCAN) [36.58] algorithms are less sensitive to outliers and can discover clusters of irregular (p-dimensional) shapes. DBSCAN is designed to discover clusters and noise in a spatial database. The advantage of DBSCAN over other clustering methods is its ability to represent specific structure in the analysis explicitly. DBSCAN has two key parameters: neighborhood size (ϵ) and minimum cluster size ($n_{\min}$). The neighborhood of an object within a radius ϵ is called the ϵ-neighborhood of the object. If the ϵ-neighborhood of an object contains at least $n_{\min}$ objects, then the object is called a core object. To find a cluster, DBSCAN starts with an arbitrary object o in the database. If the object o is a core object w.r.t. ϵ and $n_{\min}$, then a new cluster with o as the core object is created. DBSCAN continues to retrieve all density-reachable objects from the core object and add them to the cluster.

GDBSCAN [36.59] generalizes the two key parameters of the DBSCAN algorithm such that it can cluster point objects and spatially extended objects according to an arbitrarily selected combination of attributes. The neighborhood of an object is now defined by a binary predicate η on a data set that is reflexive and symmetric. If η is true, then the neighborhood of an object is called the η-neighborhood of an object. In other words, the η-neighborhood of an object is a set of objects, S, which meet the condition that η is true. Corresponding to $n_{\min}$, another predicate, $w_{\min}$ of the set of objects, S, is defined such that it is true if and only if the weighted cardinality for the set, wCard(S), is greater or equal to the minimum cardinality (MinCard), i. e. wCard(S) $\geq$ MinCard.

Finally, ordering points to identify the clustering structure (OPTICS) [36.60] is a method of cluster analysis that produces an augmented ordering of the database representing its density-based clustering structure. This method by itself does not produce a clustering of a data set explicitly. The information produced by OPTICS includes representative points, arbitrarily shaped clusters and intrinsic clustering structure, which can then be used by a clustering algorithm when selecting clustering settings. This same information can also be used by a human expert to gain insight into the clustering structure of the data.

Self-Organizing (Feature) Maps (SOMs) [36.61] belong to the class of ANNs called *unsupervised learning networks*. SOMs can be organized as a single layer or as two layers of neuron nodes. In this arrangement, the input layer consists of p nodes corresponding to the real-valued input vector of dimension p. The input layer nodes are connected to a second layer of nodes U. By means of lateral connections, the nodes in U form a lattice structure of dimensionality M. Typically M is much smaller than p. By means of a learning algorithm, the network *discovers* the clusters within the data. It is possible to alter the discovered clusters by varying the learning parameters of the network. The SOM is especially suitable for data survey because it has appealing visualization properties. It creates a set of prototype vectors representing the data set and carries out a topology-preserving projection of the prototypes from the p-dimensional input space onto a low-dimensional (typically two-dimensional) grid. This ordered grid can be used as a convenient visualization surface for showing different features of the SOM (and thus of the data), for example, the cluster structure. While the axes of such a grid do not correspond to any measurement, the spatial relationships among the clusters do correspond to relationships in p-dimensional space. Another attractive feature of the SOM is its ability to discover arbitrarily shaped clusters organized in a nonlinear space.

36.3.3 Software

Several DM software packages are available at a wide range of prices, of which six of the most popular packages are:

- SAS Enterprise Miner (www.sas.com/technologies/analytics/datamining/miner/),
- SPSS Clementine (www.spss.com/clementine/),
- XLMiner in Excel (www.xlminer.net),
- Ghostminer (www.fqspl.com.pl/ghostminer/),
- Quadstone (www.quadstone.com/),
- Insightful Miner (www.splus.com/products/iminer/).

Haughton et al. [36.62] present a review of the first five listed above. The SAS and SPSS packages have the most complete set of KDD/DM tools (data handling, DM modeling, and graphics), while Quadstone is the most limited. Insightful Miner was developed by *S+* [www.splus.com], but does not require knowledge of the *S+* language, which is only recommended for users that are familiar with statistical modeling. For statisticians, the advantage is that Insightful Miner can be integrated with more sophisticated DM methods available with *S+*, such as flexible and mixture discriminant analysis. All six packages include trees and clustering, and all except Quadstone include ANN modeling. The SAS, SPSS, and XLMiner packages include discriminant analysis and association rules. Ghostminer is the only one that offers SVM tools.

SAS, SPSS, and Quadstone are the most expensive (over $ 40 000) while XLMiner is a good deal for the price (under $ 2 000). The disadvantage of XLMiner is that it cannot handle very large data sets. Each package has certain specializations, and potential users must carefully investigate these choices to find the package that best fits their KDD/DM needs. Below we describe some other software options for the DM modeling methods presented.

GLM or linear models are the simplest of DM tools and most statistical software can fit them, such as SAS, SPSS, *S+*, and Statistica [www.statsoftinc.com/]. However, it should be noted that Quadstone only offers a regression tool via scorecards, which is not the same as statistical linear models. GAM requires access to more sophisticated statistical software, such as *S+*.

Software for CART, MART, and MARS is available from Salford Systems [www.salford-systems.com]. SAS Enterprise Miner includes CHAID, CART, and the machine learning program C4.5 [www.rulequest.com] [36.63], which uses classifiers to generate decision trees and if–then rules. SPSS Clementine and Insightful Miner also include CART, but Ghostminer and XLMiner utilize different variants of decision trees. QUEST [www.stat.wisc.edu/~loh/quest.html] is available in SPSS's AnswerTree software and Statistica.

Although ANN software is widely available, the most complete package is Matlab's [www.mathworks.com] Neural Network Toolbox. Information on SVM software is available at [www.support-vector.net/software.html]. One good option is Matlab's SVM Toolbox.

36.4 DM Research and Applications

Many industrial and business applications require modeling and monitoring processes with real-time data of different types: real values, categorical, and even text. DM is an effective tool for extracting process knowledge and discovering data patterns to provide a control aid for these processes. Advanced DM research involves complex system modeling of heterogeneous objects, where adaptive algorithms are necessary to capture dynamic system behavior.

36.4.1 Activity Monitoring

One important DM application is the development of an effective data modeling and monitoring system for understanding customer profiles and detecting fraudulent behavior. This is generally referred to as *activity monitoring for interesting events requiring action* [36.64]. Other activity monitoring examples include credit card or insurance fraud detection, computer intrusion detection, some forms of fault detection, network performance monitoring, and news story monitoring.

Although activity monitoring has only recently received attention in the information industries, solutions to similar problems were developed long ago in the manufacturing industries, under the moniker *statistical process control* (SPC). SPC techniques have been used routinely for online process control and monitoring to achieve process stability and to improve process capability through variation reduction. In general, all processes are subject to some natural variability regardless of their state. This natural variability is usually small and unavoidable and is referred to as *common cause variation*. At the same time, processes may be subject to other variability caused by improper machine adjustment, operator errors, or low-quality raw material. This variability is usually large, but avoidable, and is referred to as *special cause variation*. The basic objective of SPC is to detect the occurrence of special cause variation (or process shifts) quickly, so that the process can be investigated and corrective action may be taken before quality deteriorates and defective units are produced. The main ideas and methods of SPC were developed in the 1920s by Walter Shewhart of Bell Telephone Laboratories and have had tremendous success in manufacturing applications [36.65, 66]. *Montgomery* and *Woodall* [36.67] provide a comprehensive panel discussion on SPC, and multivariate methods are reviewed by *Hayter* and *Tsui* [36.68] and *Mason* et al. [36.69].

Although the principle of SPC can be applied to service industries, such as business process monitoring, fewer applications exist for two basic reasons that *Montgomery* [36.65] identified. First, the system that needs to be monitored and improved is obvious in manufacturing applications, while it is often difficult to define and observe in service industries. Second, even if the system can be clearly specified, most non-manufacturing operations do not have natural measurement systems that reflect the performance of the system. However, these obstacles no longer exist, due to the many natural and advanced measurement systems that have been developed. In the telecommunications industry, for example, advanced software and hardware technologies make it possible to record and process huge amounts of daily data in business transactions and service activities. These databases contain potentially useful information to the company that may not be discovered without knowledge extraction or DM tools.

While SPC ideas can be applied to business data, SPC methods are not directly applicable. Existing SPC theories are based on small or medium-sized samples, and the basic hypothesis testing approach is intended to detect only simple shifts in a process mean or variance. Recently, *Jiang* et al. [36.70] successfully generalized the SPC framework to model and track thousands of diversified customer behaviors in the telecommunication industry. The challenge is to develop an integrated strategy to monitor the performance of an entire multi-stage system and to develop effective and efficient techniques for detecting the systematic changes that require action.

A dynamic business process can be described by the dynamic linear models introduced by *West* [36.71],

$$\begin{aligned} &\textit{Observation equation}: X_t = A_t\theta_t + \Delta_t \,, \\ &\textit{System evolution equation}: \theta_t = B_t\theta_{t-1} + \Lambda_t \,, \\ &\textit{Initial information}: \pi(S_0) \,, \end{aligned}$$

where A_t and B_t represent observation and state transition matrices, respectively, and Δ_t and Λ_t represent observation and system transition errors, respectively. Based on the dynamic system model, a model-based process monitoring and root-cause identification method can be developed. Monitoring and diagnosis includes fault pattern generation and feature extraction, isolation of the critical processes, and root-cause identification. *Jiang* et al. [36.70] utilize this for individual customer prediction and monitoring. In general, individual modeling is computationally intractable and

cluster models should be developed with mixture distributions [36.72].

One particularly competitive industry is telecommunications. Since divestiture and government deregulation, various telephone services, such as cellular, local and long distance, domestic and commercial, have become battle grounds for telecommunication service providers. Because of the data and information oriented nature of the industry, DM methods for knowledge extraction are critical. To remain competitive, it is important for companies to develop business planning systems that help managers make good decisions. In particular, these systems will allow sales and marketing people to establish successful customer loyalty programs for churn prevention and to develop fraud detection modules for reducing revenue loss through market segmentation and customer profiling.

A major task in this research is to develop and implement DM tools within the business planning system. The objectives are to provide guidance for targeting business growth, to forecast year-end usage volume and revenue growth, and to value risks associated with the business plan periodically. Telecommunication business services include voice and non-voice services, which can be further categorized to include domestic, local, international, products, toll-free calls, and calling cards. For usage forecasting, a minutes growth model is utilized to forecast domestic voice usage. For revenue forecasting, the average revenue per minute on a log scale is used as a performance measure and is forecasted by a double exponential smoothing growth function. A structural model is designed to decompose the business growth process into three major subprocesses: add, disconnect, and base. To improve explanatory power, the revenue unit is further divided into different customer groups. To compute confidence and prediction intervals, bootstrapping and simulation methods are used.

To understand the day effect and seasonal effect, the concept of bill-month equivalent business days (EBD) is defined and estimated. To estimate EBD, the factor characteristics of holidays (non-EBD) are identified and eliminated and the day effect is estimated. For seasonality, the US Bureau of the Census X-11 seasonal adjustment procedure is used.

36.4.2 Mahalanobis–Taguchi System

Genichi Taguchi is best known for his work on robust design and design of experiments. The Taguchi robust design methods have generated a considerable amount of discussion and controversy and are widely used in manufacturing [36.73–77]. The general consensus among statisticians seems to be that, while many of Taguchi's overall ideas on experimental design are very important and influential, the techniques he proposed are not necessarily the most effective statistical methods. Nevertheless, Taguchi has made significant contributions in the area of quality control and quality engineering. For DM, Taguchi has recently popularized the *Mahalanobis–Taguchi System* (MTS), a new set of tools for diagnosis, classification, and variable selection. The method is based on a Mahalanobis distance scale that is utilized to measure the level of abnormality in *abnormal* items as compared to a group of *normal* items. First, it must be demonstrated that a Mahalanobis distance measure based on all available variables is able to separate the abnormal from the normal items. Should this be successfully achieved, orthogonal arrays and signal-to-noise ratios are used to select an *optimal* combination of variables for calculating the Mahalanobis distances.

The MTS method has been claimed to be very powerful for solving a wide range of problems, including manufacturing inspection and sensing, medical diagnosis, face and voice recognition, weather forecasting, credit scoring, fire detection, earthquake forecasting, and university admissions. Two recent books have been published on the MTS method by *Taguchi* et al. [36.78] and *Taguchi* and *Jugulum* [36.79]. Many successful case studies in MTS have been reported in engineering and science applications in many large companies, such as Nissan Motor Co., Mitsubishi Space Software Co., Xerox, Delphi Automotive Systems, ITT Industries, Ford Motor Company, Fuji Photo Film Company, and others. While the method is getting a lot of attention in many industries, very little research [36.80] has been conducted to investigate how and when the method is appropriate.

36.4.3 Manufacturing Process Modeling

One area of DM research in manufacturing industries is quality and productivity improvement through DM and knowledge discovery. Manufacturing systems nowadays are often very complicated and involve many manufacturing process stages where hundreds or thousands of in-process measurements are taken to indicate or initiate process control of the system. For example, a modern semiconductor manufacturing process typically consists of over 300 steps, and in each step, multiple pieces of equipment are used

to process the wafer. Inappropriate understanding of interactions among in-process variables will create inefficiencies at all phases of manufacturing, leading to long product/process realization cycle times and long development times, and resulting in excessive system costs.

Current approaches to DM in electronics manufacturing include neural networks, decision trees, Bayesian models and rough set theory [36.81, 82]. Each of these approaches carries certain advantages and disadvantages. Decision trees, for instance, produce intelligible rules and hence are very appropriate for generating process control or design of experiments strategies. They are, however, generally prone to outlier and imperfect data influences. Neural networks, on the other hand, are robust against data abnormalities but do not produce readily intelligible knowledge. These methods also differ in their ability to handle high-dimensional data, to discover arbitrarily shaped clusters [36.58] and to provide a basis for intuitive visualization [36.83]. They can also be sensitive to training and model building parameters [36.60]. Finally, the existing approaches do not take into consideration the localization of process parameters. The patterns or clusters identified by existing approaches may include parameters from a diverse set of components in the system. Therefore, a combination of methods that complement each other to provide a complete set of desirable features is necessary.

It is crucial to understand process structure and yield components in manufacturing, so that problem localization can permit reduced production costs. For example, semiconducture manufacturing practice shows that over 70% of all fatal detects and close to 90% of yield excursions are caused by problems related to process equipment [36.84]. Systematic defects can be attributed to many categories that are generally associated with technologies and combinations of different process operations. To implement DM methods successfully for knowledge discovery, some future research for manufacturing process control must include yield modeling, defect modeling and variation propagation.

Yield Modeling

In electronics manufacturing, the ANSI standards [36.85] and practice gene rally assume that the number of defects on an electronics product follows a Poisson distribution with mean λ. The Poisson random variable is an approximation of the sum of independent Bernoulli trials, but defects on different components may be correlated since process yield critically depends on product groups, process steps, and types of defects [36.86]. Unlike traditional defect models, an appropriate logit model can be developed as follows. Let the number of defects of category X on an electronics product be

$$U_X = \sum Y_X$$

and

$$\text{logit}\big[E(Y_X)\big] = \alpha_X^0 + \alpha_X^O \cdot O_X + \alpha_X^C \cdot C_X + \alpha_X^{OC} \cdot O_X \cdot C_X \,,$$

where $\text{logit}(z) = \log[z/(1-z)]$ is the link function for Bernoulli distributions, and Y_X is a Bernoulli random variable representing a defect from defect category X. The default logit of the failure probability is α_X^0, and α_X^O and α_X^C are the main effects of operations (O_X) and components (C_X). Since the Y_Xs are correlated, this model will provide more detailed information about defects.

Multivariate Defect Modeling

Since different types of defects may be caused by the same operations, multivariate Poisson models are necessary to account for correlations among different types of defects. The *trivariate reduction method* suggests an additive Poisson model for the vector of Poisson counts $\boldsymbol{U} = (U_1, U_2, \cdots, U_k)'$,

$$\boldsymbol{U} = \boldsymbol{AV} \,,$$

where $\boldsymbol{A}$ is a matrix of zeros and ones, and $\boldsymbol{V} = (v_1, v_2, \cdots, v_p)'$ consists of independent Poisson variables v_i. The variance–covariance matrix takes the form $\text{Var}(U) = A\Sigma A' = \boldsymbol{\Phi} + \nu\nu'$, where $\boldsymbol{\Phi} = \text{diag}(\mu_i)$ is a diagonal matrix with the mean of the individual series, and ν is the common covariance term. Note that the v_i are essentially latent variables, and a factor analysis model can be developed for analyzing multivariate discrete Poisson variables such that

$$\log[E(\boldsymbol{U})] = \mu + \boldsymbol{L} \cdot \boldsymbol{F} \,,$$

where $\boldsymbol{U}$ is the vector of defects, $\boldsymbol{L}$ is the matrix of factor loadings, and $\boldsymbol{F}$ contains common factors representing effects of specific operations. By using factor analysis, it is possible to relate product defects to the associated packages and operations.

Multistage Variation Propagation

Inspection tests in an assembly line usually have functional overlap, and defects from successive inspection

stations exhibit strong correlations. Modeling serially correlated defect counts is an important task for defect localization and yield prediction. Poisson regression models, such as the generalized event-count method [36.87] and its alternatives, can be utilized to account for serial correlations of defects in different inspection stations. Factor analysis methods based on hidden Markov models [36.88] can also be constructed to investigate how variations are propagated through assembly lines.

36.5 Concluding Remarks

While DM and KDD methods are gaining recognition and have become very popular in many companies and enterprises, the success of these methods is still somewhat limited. Below, we discuss a few obstacles.

First, the success of DM depends on a close collaboration of subject-matter experts and data modelers. In practice, it is often easy to identify the right subject-matter expert, but difficult to find the qualified data modeler. While the data modeler must be knowledgeable and familiar with DM methods, it is more important to be able to formulate real problems such that the existing methods can be applied. In reality, traditional academic training mainly focuses on knowledge of modeling algorithms and lacks training in problem formulation and interpretation of results. Consequently, many modelers are very efficient in fitting models and algorithms to data, but have a hard time determining when and why they should use certain algorithms. Similarly, the existing commercial DM software systems include many sophisticated algorithms, but lack of guidance on which algorithms to use.

Second, implementation of DM is difficult to apply effectively across an industry. Although it is clear that extracting hidden knowledge and trends across an industry would be useful and beneficial to all companies in the industry, it is typically impossible to integrate the detailed data from competing companies due to confidentiality and proprietary issues. Currently, the industry practice is that each company will integrate their own detailed data with the more general, aggregated industry-wide data for knowledge extraction. It is obvious that this approach will be significantly less effective than the approach of integrating the detailed data from all competing companies. It is expected that, if these obstacles can be overcome, the impact of the DM and KDD methods will be much more prominent in industrial and commercial applications.

References

36.1 M. J. A. Berry, G. Linoff: *Mastering Data Mining: The Art and Science of Customer Relationship Management* (Wiley, New York 2000)

36.2 E. Wegman: Data Mining Tutorial, Short Course Notes, Interface 2001 Symposium, Cosa Mesa, Californien (2001)

36.3 P. Adriaans, D. Zantinge: *Data Mining* (Addison-Wesley, New York 1996)

36.4 J. H. Friedman: *Data Mining and Statistics: What is the Connection? Technical Report* (Stat. Dep., Stanford University 1997)

36.5 K. B. Clark, T. Fujimoto: Product Development and Competitiveness, J. Jpn Int. Econ. **6(2)**, 101–143 (1992)

36.6 D. W. LaBahn, A. Ali, R. Krapfel: New Product Development Cycle Time. The Influence of Project and Process Factors in Small Manufacturing Companies, J. Business Res. **36(2)**, 179–188 (1996)

36.7 J. Han, M. Kamber: *Data Mining: Concept and Techniques* (Morgan Kaufmann, San Francisco 2001)

36.8 T. Hastie, J. H. Friedman, R. Tibshirani: *Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, Berlin Heidelberg New York 2001)

36.9 S. Weisberg: *Applied Linear Regression* (Wiley, New York 1980)

36.10 G. Seber: *Multivariate Observations* (Wiley, New York 1984)

36.11 J. Neter, M. H. Kutner, C. J. Nachtsheim, W. Wasserman: *Applied Linear Statistical Models*, 4th edn. (Irwin, Chicago 1996)

36.12 A. E. Hoerl, R. Kennard: Ridge Regression: Biased Estimation of Nonorthogonal Problems, Technometrics **12**, 55–67 (1970)

36.13 R. Tibshirani: Regression Shrinkage and Selection via the Lasso, J. R. Stat. Soc. Series B **58**, 267–288 (1996)

36.14 A. Agresti: *An Introduction to Categorical Data Analysis* (Wiley, New York 1996)

36.15 D. Hand: *Discrimination and Classification* (Wiley, Chichester 1981)

36.16 P. McCullagh, J. A. Nelder: *Generalized Linear Models*, 2nd edn. (Chapman Hall, New York 1989)
36.17 T. Hastie, R. Tibshirani: *Generalized Additive Models* (Chapman Hall, New York 1990)
36.18 W. S. Cleveland: Robust Locally-Weighted Regression and Smoothing Scatterplots, J. Am. Stat. Assoc. **74**, 829–836 (1979)
36.19 R. L. Eubank: *Spline Smoothing and Nonparametric Regression* (Marcel Dekker, New York 1988)
36.20 G. Wahba: *Spline Models for Observational Data*, Applied Mathematics, Vol. 59 (SIAM, Philadelphia 1990)
36.21 W. Härdle: *Applied Non-parametric Regression* (Cambridge Univ. Press, Cambridge 1990)
36.22 D. Biggs, B. deVille, E. Suen: A Method of Choosing Multiway Partitions for Classification and Decision Trees, J. Appl. Stat. **18(1)**, 49–62 (1991)
36.23 B. D. Ripley: *Pattern Recognition and Neural Networks* (Cambridge Univ. Press, Cambridge 1996)
36.24 L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone: *Classification and Regression Trees* (Wadsworth, Belmont, California 1984)
36.25 J. N. Morgan, J. A. Sonquist: Problems in the Analysis of Survey Data, and a Proposal, J. Am. Stat. Assoc. **58**, 415–434 (1963)
36.26 A. Fielding: Binary segmentation: The Automatic Interaction Detector and Related Techniques for Exploring Data Structure. In: *The Analysis of Survey Data, Volume I: Exploring Data Structures*, ed. by C. A. O'Muircheartaigh, C. Payne (Wiley, New York 1977) pp. 221–258
36.27 W. Y. Loh, N. Vanichsetakul: Tree-Structured Classification Via Generalized Discriminant Analysis, J. Am. Stat. Assoc. **83**, 715–728 (1988)
36.28 W. D. Lo. Chaudhuri, W. Y. Loh, C. C. Yang: Generalized Regression Trees, Stat. Sin. **5**, 643–666 (1995)
36.29 W. Y. Loh, Y. S. Shih: Split-Selection Methods for Classification Trees, Statistica Sinica **7**, 815–840 (1997)
36.30 J. H. Friedman, T. Hastie, R. Tibshirani: Additive Logistic Regression: a Statistical View of Boosting, Ann. Stat. **28**, 337–407 (2000)
36.31 Y. Freund, R. Schapire: Experiments with a New Boosting Algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, Bari, Italy 1996, ed. by M. Kaufmann, (Bari, Italy 1996) 148–156
36.32 L. Breiman: Bagging Predictors, Machine Learning **26**, 123–140 (1996)
36.33 J. H. Friedman: Greedy Function Approximation: a Gradient Boosting Machine, Ann. Stat. **29**, 1189–1232 (2001)
36.34 J. H. Friedman: Stochastic Gradient Boosting, Computational Statistics and Data Analysis **38(4)**, 367–378 (2002)
36.35 J. H. Friedman: Multivariate Adaptive Regression Splines (with Discussion), Ann. Stat. **19**, 1–141 (1991)
36.36 J. H. Friedman, B. W. Silverman: Flexible Parsimonious Smoothing and Additive Modeling, Technometrics **31**, 3–39 (1989)
36.37 R. P. Lippmann: An Introduction to Computing with Neural Nets, IEEE ASSP Magazine **April**, 4–22 (1987)
36.38 S. S. Haykin: *Neural Networks: A Comprehensive Foundation*, 2nd edn. (Prentice Hall, Upper Saddle River 1999)
36.39 H. White: Learning in Neural Networks: a Statistical Perspective, Neural Computation **1**, 425–464 (1989)
36.40 A. R. Barron, R. L. Barron, E. J. Wegman: Statistical Learning Networks: A Unifying View, Computer Science and Statistics: Proceedings of the 20th Symposium on the Interface 1992, ed. by E. J. Wegman, D. T. Gantz, J. J. Miller (American Statistical Association, Alexandria, VA 1992) 192–203
36.41 B. Cheng, D. M. Titterington: Neural Networks: A Review from a Statistical Perspective (with discussion), Stat. Sci **9**, 2–54 (1994)
36.42 D. Rumelhart, G. Hinton, R. Williams: Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. 1: Foundations*, ed. by D. E. Rumelhart, J. L. McClelland (MIT, Cambridge 1986) pp. 318–362
36.43 V. Vapnik: *The Nature of Statistical Learning* (Springer, Berlin Heidelberg New York 1996)
36.44 C. J. C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Knowledge Discovery and Data Mining **2(2)**, 121–167 (1998)
36.45 J. Shawe-Taylor, N. Cristianini: *Kernel Methods for Pattern Analysis* (Cambridge Univ. Press, Cambridge 2004)
36.46 N. Cristianini, J. Shawe-Taylor: *An Introduction to Support Vector Machines* (Cambridge Univ. Press, Cambridge 2000)
36.47 P. Huber: Ann. Math. Stat., Robust Estimation of a Location Parameter **53**, 73–101 (1964)
36.48 B. V. Dasarathy: *Nearest Neighbor Pattern Classification Techniques* (IEEE Computer Society, New York 1991)
36.49 T. Hastie, R. Tibshirani: Discriminant Adaptive Nearest-Neighbor Classification, IEEE Pattern Recognition and Machine Intelligence **18**, 607–616 (1996)
36.50 T. Hastie, R. Tibshirani, A. Buja: Flexible Discriminant and Mixture Models. In: *Statistics and Artificial Neural Networks*, ed. by J. Kay, M. Titterington (Oxford Univ. Press, Oxford 1998)
36.51 J. R. Koza: *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT, Cambridge 1992)
36.52 W. Banzhaf, P. Nordin, R. E. Keller, F. D. Francone: *Genetic Programming: An Introduction* (Morgan Kaufmann, San Francisco 1998)
36.53 P. W. H. Smith: Genetic Programming as a Data-Mining Tool. In: *Data Mining: A Heuristic Approach*,

ed. by H. A. Abbass, R. A. Sarker, C. S. Newton (Idea Group Publishing, London 2002) pp. 157–173
36.54 R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo: *Fast Discovery Of Association Rules: Advances in Knowledge Discovery and Data Mining* (MIT, Cambridge 1995) Chap. 12
36.55 A. Gordon: *Classification*, 2nd edn. (Chapman Hall, New York 1999)
36.56 J. A. Hartigan, M. A. Wong: A *K*-Means Clustering Algorithm, Appl. Stat. **28**, 100–108 (1979)
36.57 L. Kaufman, P. Rousseeuw: *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley, New York 1990)
36.58 M. Ester, H.-P. Kriegel, J. Sander, X. Xu: A Density-Based Algorithm for Discovering Cluster in Large Spatial Databases, Proceedings of 1996 International Conference on Knowledge Discovery and Data Mining (KDD96), Portland 1996, ed. by E. Simoudis, J. Han, U. Fayyad (AAAI Press, Menlo Park 1996) 226–231
36.59 J. Sander, M. Ester, H.-P. Kriegel, X. Xu: Density-Based Clustering in Spatial Databases: The Algorithm DGBSCAN and its Applications, Data Mining and Knowledge Discovery **2(2)**, 169–194 (1998)
36.60 M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander: OPTICS: Ordering Points to Identify the Clustering Structure, Proc. ACMSIGMOD Int. Conf. on Management of Data, Philadelphia, Pennsylvania June 1-3, 1999 (ACM Press, New York 1999) 49–60
36.61 T. Kohonen: *Self-Organization and Associative Memory*, 3rd edn. (Springer, Berlin Heidelberg New York 1989)
36.62 D. Haughton, J. Deichmann, A. Eshghi, S. Sayek, N. Teebagy, H. Topi: A Review of Software Packages for Data Mining, Amer. Stat. **57(4)**, 290–309 (2003)
36.63 J. R. Quinlan: *C4.5: Programs for Machine Learning* (Morgan Kaufmann, San Mateo 1993)
36.64 T. Fawcett, F. Provost: Activity Monitoring: Noticing Interesting Changes in Behavior, Proceedings of KDD-99, San Diego 1999, (San Diego, CA 1999) 53–62
36.65 D. C. Montgomery: *Introduction to Statistical Quality Control*, 5th edn. (Wiley, New York 2001)
36.66 W. H. Woodall, K.-L. Tsui, G. R. Tucker: A Review of Statistical and Fuzzy Quality Control Based on Categorical Data, Frontiers in Statistical Quality Control **5**, 83–89 (1997)
36.67 D. C. Montgomery, W. H. Woodall: A Discussion on Statistically-Based Process Monitoring and Control, J. Qual. Technol. **29**, 121–162 (1997)
36.68 A. J. Hayter, K.-L. Tsui: Identification and Qualification in Multivariate Quality Control Problems, J. Qual. Tech. **26(3)**, 197–208 (1994)
36.69 R. L. Mason, C. W. Champ, N. D. Tracy, S. J. Wierda, J. C. Young: Assessment of Multivariate Process Control Techniques, J. Qual. Technol. **29**, 140–143 (1997)
36.70 W. Jiang, S.-T. Au, K.-L. Tsui: A Statistical Process Control Approach for Customer Activity Monitoring, Technical Report, AT&T Labs (2004)
36.71 M. West, J. Harrison: *Bayesian Forecasting and Dynamic Models*, 2nd edn. (Springer, New York 1997)
36.72 C. Fraley, A. E. Raftery: Model-based Clustering, Discriminant Analysis, and Density Estimation, J. Amer. Stat. Assoc. **97**, 611–631 (2002)
36.73 G. Taguchi: *Introduction to Quality Engineering: Designing Quality into Products and Processes* (Asian Productivity Organization, Tokyo 1986)
36.74 G. E. P. Box, R. N. Kacker, V. N. Nair, M. S. Phadke, A. C. Shoemaker, C. F. Wu: Quality Practices in Japan, Qual. Progress **March**, 21–29 (1988)
36.75 V. N. Nair: Taguchi's Parameter Design: A Panel Discussion, Technometrics **34**, 127–161 (1992)
36.76 K.-L. Tsui: An Overview of Taguchi Method and Newly Developed Statistical Methods for Robust Design, IIE Trans. **24**, 44–57 (1992)
36.77 K.-L. Tsui: A Critical Look at Taguchi's Modeling Approach for Robust Design, J. Appl. Stat. **23**, 81–95 (1996)
36.78 G. Taguchi, S. Chowdhury, Y. Wu: *The Mahalanobis–Taguchi System* (McGraw-Hill, New York 2001)
36.79 G. Taguchi, R. Jugulum: *The Mahalanobis–Taguchi Strategy: A Pattern Technology System* (Wiley, New York 2002)
36.80 W. H. Woodall, R. Koudelik, K.-L. Tsui, S. B. Kim, Z. G. Stoumbos, C. P. Carvounis: A Review and Analysis of the Mahalanobis–Taguchi System, Technometrics **45(1)**, 1–15 (2003)
36.81 A. Kusiak, C. Kurasek: Data Mining of Printed-Circuit Board Defects, IEEE Transactions on Robotics and Automation **17(2)**, 191–196 (2001)
36.82 A. Kusiak: Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing, IEEE Transactions on Electronics Packaging Manufacturing **24(1)**, 44–50 (2001)
36.83 A. Ultsch: *Information and Classification: Concepts, Methods and Applications* (Springer, Berlin Heidelberg New York 1993)
36.84 A. Y. Wong: A Statistical Approach to Identify Semiconductor Process Equipment Related Yield Problems, IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Paris 1997 (IEEE Computer Society, Paris, France 1997) 20–22
36.85 ANSI (2002). Am. Nat. Standards Institute, IPC-9261, In-Process DPMO and Estimated Yield for PWB
36.86 M. Baron, C. K. Lakshminarayan, Z. Chen: Markov Random Fields In Pattern Recognition For Semiconductor Manufacturing, Technometrics **43**, 66–72 (2001)
36.87 G. King: Event Count Models for International Relations: Generalizations and Applications, International Studies Quarterly **33(2)**, 123–147 (1989)
36.88 P. Smyth: Hidden Markov models for fault detection in dynamic systems, Pattern Recognition **27(1)**, 149–164 (1994)

Part E Modeling and Simulation Methods

Part E contains ten chapters and focuses on statistical methods and modeling. Chapt. 37 provides an overview of several well-known bootstrap methods, including Efron's bootstrap and Studentized bootstrap interval for constructing confidence intervals and introduces some recently developed bootstrap methods such as the estimation-function bootstrap and the Markov-chain marginal bootstrap. Chapter 38 discusses generalized linear mixed models for correlated non-normal data and various methods for random-effect model parameters including the EM algorithms, penalized quasi-likelihood, the Markov-chain Newton–Raphson, the stochastic approximation, and the S–U algorithm. Chapter 39 focuses on the design and analysis of cluster randomized trials. This chapter also describes cost-efficiency models with covariates and the robustness of optimal designs, including both the number of clusters and cluster size. Chapter 40 discuss a semiparametric estimation method for an extension of the semiparametric regression model, called the two-way semilinear model, for normalization to estimate normalization curves and its applications to microarray data. Chapter 41 covers the development of latent-variable models for longitudinal data such as the generalized linear latent and mixed model, hierarchical latent-variable models, the linear mixed model for multivariate longitudinal responses as well as structural-equation models with latent variables for longitudinal data.

The next two chapters focus on genetic algorithms and scan statistics. Chapter 42 provides an overview of the concept of genetic algorithms, including hybrid genetic algorithms, adaptive genetic algorithms and fuzzy-logic controllers, and their applications in scheduling problems, network design, reliability design-optimization problems, logistic network, and transportation-related problems. Chapter 43 describes the concepts of scan statistics and the various types used to localize large clusters in continuous time, space, and on a two-dimensional lattice. It also discusses recent double-scan statistics methods that allow practitioners to test for some unusual lagged clustering of different types of events and complex systems.

The final three chapters focus on various issues in maintenance modeling. Chapter 44 describes a condition-based failure-prediction method consisting of both a computer simulation and an experiment on a DC motor for preventive maintenance using the Kalman filter. The applications of the method and experimental set ups with related system parameters and experimental results are also discussed. Chapter 45 gives a brief introduction to maintenance modeling and discusses generalized multistate maintenance models for repairable systems as well as condition-based inspection strategies for degraded systems with multiple, competing failure processes such as degradation processes and random shocks, while Chapt. 46 presents a review of major maintenance models and policies in the maintenance literature that are commonly used in practice and discusses various recent maintenance models with consideration of repair policies and inspection with human errors.

37. Bootstrap, Markov Chain and Estimating Function

In this chapter, we first review bootstrap methods for constructing confidence intervals (regions). We then discuss the following three important properties of these methods: (i) invariance under reparameterization; (ii) automatic computation; and (iii) higher order accuracy. The greatest potential value of the bootstrap lies in complex situations, such as nonlinear regression or high-dimensional parameters for example. It is important to have bootstrap procedures that can be applied to these complex situations, but still have the three desired properties. The main purpose of this chapter is to introduce two recently developed bootstrap methods: the estimation function bootstrap, and the Markov chain marginal bootstrap method. The estimating function bootstrap has all three desired properties and it can also be applied to complex situations. The Markov chain marginal bootstrap is designed for high-dimensional parameters.

37.1 Overview

For a statistical model involving an unknown parameter θ, the two main statistical inference issues are usually: (i) point estimation $\hat{\theta}$ (how to estimate the unknown parameter θ); and (ii) how to assess the accuracy of this estimator $\hat{\theta}$ (in terms of the standard deviation or confidence interval of the unknown θ). Statisticians usually try to find the exact distribution or asymptotic distribution of the estimator $\hat{\theta}$. However, it is difficult to obtain the exact distribution or asymptotic distribution in a lot of situations. Sometimes the asymptotic distribution can be obtained, but the distribution of $\hat{\theta}$ is not well approximated. Bootstrap provides a general methodology for constructing confidence intervals for unknown parameters.

In this chapter, we first discuss bootstrap methods in terms of the following three important properties: (1) invariance under reparameterization, (2) automatic computation, and (3) higher order accuracy. To illustrate these three properties, let's consider the following simple model. Assume that $Y_1, \ldots, Y_n$ is a random sample from some unknown distribution F. Let $y = (y_1, \ldots, y_n)$ be the realization of Y. Suppose that $\theta = \theta(F)$ is the unknown parameter of interest. This θ could be the mean, or variance, or some other function of the distribution F. Let $\hat{\theta} = \hat{\theta}(y_1, \ldots, y_n)$ be the estimator of θ based on the observation y.

37.1.1 Invariance under Reparameterization

Suppose θ is a scale parameter and that $h(\theta)$ is a strictly monotonic function in the parameter space of θ. Then the new reparameterized statistical model is based on $(Y_1, \ldots, Y_n)$ and the parameter $\xi = h(\theta)$. Suppose that an estimation procedure gives $\hat{\theta}$ as the estimator of θ based on $(Y_1, \ldots, Y_n)$ and parameter θ, and $\hat{\xi}$ as the

estimator of ξ based on $(Y_1, \ldots, Y_n)$ and parameter ξ respectively. This procedure is said to be *invariant* under reparameterization if $\hat{\xi} = h(\hat{\theta})$. It is well known that the maximum likelihood procedure is invariant under reparameterization. But the moment estimation procedure is usually not invariant under reparameterization. A confidence interval procedure is *invariant* under reparameterization, if $[\hat{\theta}(\alpha/2), \hat{\theta}(1-\alpha/2)]$ is the $1-\alpha$ level confidence interval of θ based on this procedure, and $[h(\hat{\theta}[\alpha/2]), h(\hat{\theta}[1-\alpha/2])]$ is the $1-\alpha$ level confidence interval of ξ based on this procedure. Here we assume that $h(\theta)$ is a strictly increasing function. When a procedure is not invariant under reparameterization, it is usually very important to select a good transformation and perform statistical inference after transformation. This has been a topic of research in classical statistics.

37.1.2 Automatic Computation

One of the most important advantages of the bootstrap method is its automatic computation; in other words it does not depend on theoretical calculation. A procedure is called an "automatic computation" if it does not depend on any extra analytical inference. In many applications, it is very difficult (sometimes impossible) to perform analytical calculations.

37.1.3 First and Higher Order Accuracy

Suppose that θ is a scale parameter and that $\hat{\theta}[\alpha]$ is the α confidence limit of θ, based on a certain procedure. Then the procedure is said to be *first-order accurate* if $P\big(\theta < \hat{\theta}[\alpha]\big) = \alpha + O(n^{-1/2})$. It is *second-order-accurate* if $P\big(\theta < \hat{\theta}[\alpha]\big) = \alpha + O(n^{-1})$. Higher order accuracy is defined in the same way. It is well known that the standard confidence interval,

$$\left[\hat{\theta} - \hat{\sigma} z_{(\alpha/2)}\,,\ \hat{\theta} + \hat{\sigma} z_{(\alpha/2)}\right],$$

is only first-order-accurate under some conditions. Here $z_{(\alpha/2)}$ is defined by $P(Z \geq z_{(\alpha)}) = \alpha$ for a standard normal random variable Z, while $\hat{\sigma}$ is an estimator of the standard deviation of $\hat{\theta}$. One advantage of using the bootstrap method is getting confidence intervals that are accurate to the second order.

In Sect. 37.2 we will introduce Efron's bootstrap for iid samples. To construct a second-order-accurate confidence interval, four different bootstrap methods are reviewed and discussed in terms of the three properties described above. We then consider three bootstrap methods for a linear model and discuss their properties.

In some more complex situations, the observations could be heteroscedastic; in other words the variances of Y_i are different. It is important to have a bootstrap procedure that remains consistent under heteroscedasticity. When θ is a high-dimensional vector it is usually more difficult to apply a bootstrap procedure because it is: (i) computational intensive; (ii) difficult to construct a good confidence region. For high dimension problems, it is often important to have reliable computational results. Some new bootstrap methods have been proposed for these complex situations.

The main propose of this chapter is to introduce some recent developments in bootstrap methodology. We consider two bootstrap methods. The first is the estimating function (EF) bootstrap proposed in *Hu* and *Kalbfleisch* [37.1]. Instead of resampling the data itself, the EF bootstrap resamples some functions of the data in order to achieve robustness to heteroscedasticity. This EF bootstrap is often the simplest computationally and it is straightforward to define studentized versions of the EF bootstrap which are invariant under reparameterization and require very little additional calculation. This method can be used to get confidence regions that are accurate to higher orders for multidimensional parameters. When the estimating function is differentiable, it can be easily extended to deal with nuisance parameter problems. Another method is called the Markov chain marginal bootstrap (MCMB), which is useful for constructing confidence intervals or regions for high dimension parameters. The MCMB is different from most bootstrap methods in two aspects: first, it solves only one-dimensional equations for a problem with any number of dimensions; second, it produces a Markov chain rather than a (conditionally) independent sequence.

In Sect. 37.3, we introduce the EF bootstrap and discuss the properties of the EF bootstrap. Some examples are used to illustrate the procedure. The Markov chain marginal bootstrap method is introduced in Sect. 37.4. In Sect. 37.5, we use the simple linear model to illustrate the MCMB algorithm and its properties. We also apply the EF bootstrap and MCMB method to different examples. In Sect. 37.6, we discuss some issues with using bootstrap methods.

37.2 Classical Bootstrap

37.2.1 Efron's Bootstrap

To start, let's consider the simplest case. Assume that $Y_1, \ldots, Y_n$ is a random sample from some unknown distribution F. Let $y = (y_1, \ldots, y_n)$ be the realization of Y. Suppose that $\theta = \theta(F)$ is the unknown parameter (scale) of interest. This θ could be the mean, the variance, or some other function of the distribution F. Let $\hat{\theta} = \hat{\theta}(y_1, \ldots, y_n)$ be the estimator of θ based on the observation y.

The main statistical goal is to find the distribution of $\hat{\theta} - \theta$. If we know this distribution, then we can do all kinds of statistical inference, (including deriving standard deviations and confidence intervals). Before Efron's bootstrap paper [37.2], researchers focused on finding the exact distribution or asymptotical distribution based on different theoretical approaches. Most of these methods depended on certain assumptions for the distribution F.

Efron's basic idea was to use computer simulation to investigate the distribution of $\hat{\theta} - \theta$. If F is a given (known) distribution, then we can use Monte Carlo simulation to get the distribution of $\hat{\theta} - \theta$. When F is unknown, the best nonparametric estimator of F is the empirical distribution, $\hat{F}$, which gives a weight of $1/n$ to each y_i. The bootstrap procedure can be summarized as follows:

- [i.] Draw a bootstrap sample $z_1^*, \ldots, z_n^*$ from distribution $\hat{F}$, which is the same as drawing $z_1^*, \ldots, z_n^*$ from $(y_1, \ldots, y_n)$ with replacement;
- [ii.] Calculate the bootstrap estimator $\hat{\theta}^* = \hat{\theta}(z_1^*, \ldots, z_n^*)$;
- [iii.] Repeat steps (i) and (ii) B (the bootstrap sample size) times to get $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$. Now we define the empirical distribution:

$$\hat{G}(x) = B^{-1} \sum_{i=1}^{n} I(\hat{\theta}_i^* \le x) .$$

  Where $I(\cdot)$ is the indication function.
- [iv.] Use the empirical distribution of $\hat{\theta}^* - \hat{\theta}$ to approximate the distribution of $\hat{\theta} - \theta$.

In early work, the bootstrap estimators described above were used to estimate the bias of the estimator $\hat{\theta}$ [37.2, 3]. However, the main contribution of the bootstrap method is to provide a new way to assess the accuracy of the estimator $\hat{\theta}$. Here we discuss two main applications: (i) estimating the standard deviation of $\hat{\theta}$, and (ii) estimating the confidence interval of θ.

Standard Deviation

A commonly used measure of the accuracy of $\hat{\theta}$ is the standard deviation of $\hat{\theta}$. Based on the bootstrap estimators, we can easily calculate the bootstrap variance estimator as:

$$\hat{\sigma}^* = \left[(B-1)^{-1} \sum_{j=1}^{B} \left(\hat{\theta}_j^* - \bar{\theta}^* \right)^2 \right]^{1/2} ,$$

where $\bar{\theta}^* = B^{-1} \sum_{j=1}^{B} \hat{\theta}_j^*$. Under very general conditions, this is a consistent estimator of the true standard deviation [37.3]. The advantage of this bootstrap standard deviation, $\hat{\sigma}^*$, is that it does not depend on analytical inference. Instead, we can get it by computer simulation. When $\hat{\sigma}$ is not available, this $\hat{\sigma}^*$ can be used as the estimator of the standard deviation of $\hat{\theta}$.

Based on $\hat{\sigma}^*$, we can construct an approximate confidence interval for the unknown parameter θ as

$$\left[\hat{\theta} - \hat{\sigma}^* z_{(\alpha/2)} , \hat{\theta} + \hat{\sigma}^* z_{(\alpha/2)} \right] .$$

While this confidence interval conforms to "automatic computation", it is not "invariant under reparameterization", and is only first-order-accurate.

Confidence Interval

First we consider the following two distributions: (i) the distribution of the estimator $\hat{\theta}$,

$$H(t) = P\left[n^{-1/2} \left(\hat{\theta} - \theta \right) \le t \right] ;$$

and the corresponding distribution of the bootstrap estimator $\hat{\theta}^*$,

$$\hat{H}(t) = P\left[n^{-1/2} \left(\hat{\theta}^* - \hat{\theta} \right) \le t \right] .$$

Under certain conditions, it can be shown that [37.4]

$$\max_{t \in [-\infty, \infty]} |H(t) - \hat{H}(t)| = O_p\left(n^{1/2} \right) .$$

By using the distribution of $\hat{\theta}^* - \hat{\theta}$ to approximate the distribution of $\hat{\theta} - \theta$, we can construct the level $1 - \alpha$ confidence interval of θ as

$$\left[2\hat{\theta} - \hat{\theta}^*(1 - \alpha/2), 2\hat{\theta} - \hat{\theta}^*(\alpha/2) \right] ,$$

where $\hat{\theta}^*(\alpha)$ is the αth quantile of the bootstrap distribution $\hat{G}$. This confidence interval is also obtained via

"automatic computation", but it is not "invariant under reparameterization", and is only first-order-accurate.

Another way to construct the confidence interval is by using the distribution of $\hat{\theta}^*$ directly. The confidence interval is defined as

$$\left[\hat{\theta}^*(\alpha/2), \hat{\theta}^*(1-\alpha/2)\right] .$$

This interval is obtained via "automatic computation" and is "invariant under reparameterization", but is only first-order-accurate.

37.2.2 Second-Order-Accurate Confidence Intervals

One of the problems that has been studied the most in bootstrap literature is how to construct higher accurate bootstrap confidence intervals; see for example [37.5–8]. Here we review four commonly used methods: (i) studentized bootstrap interval; (ii) bias-corrected and accelerated method (BC_a method); (iii) approximated bootstrap confidence (ABC) interval; and (iv) prepivoting bootstrap interval. The advantages and disadvantages of these four methods are also discussed.

Studentized Bootstrap Interval

Instead of considering $n^{1/2}\left(\hat{\theta}-\theta\right)$ directly, we use the studentized statistic

$$T = \frac{n^{1/2}(\hat{\theta}-\theta)}{\hat{\sigma}(y_1, \ldots, y_n)} ,$$

where $\hat{\sigma}^2 = \hat{\sigma}^2(y_1, \ldots, y_n)$ is an estimate of the asymptotic variance $\mathrm{Var}\left(n^{1/2}\hat{\theta}\right)$. The corresponding bootstrap studentized statistic is then

$$T^* = \frac{n^{1/2}\left(\hat{\theta}^*-\hat{\theta}\right)}{\hat{\sigma}^*(z_1^*, \ldots, z_n^*)} .$$

A large number (B, say) of independent replications give the following estimated percentiles:

$$\hat{T}^{(\alpha)} = \alpha_{th} \text{ quantile of } (T^*(b), b = 1, \ldots, B) .$$

The $100\alpha_{th}$ bootstrap-t confidence endpoint $\hat{\theta}_T[\alpha]$ is then defined to be

$$\hat{\theta}_T[\alpha] = \hat{\theta} - \hat{\sigma}\hat{T}^{(1-\alpha)} .$$

Based on Edgeworth expansions of the statistics T and T^*, *Hall* [37.9] showed that

$$P(T < v) - P(T^* < v) = O_p(n^{-1}) ,$$

where the second probability is under the bootstrap distribution, so the bootstrap-t intervals are usually second-order-accurate. The advantage of this studentized bootstrap is that it is intuitive and easy to understand.

But this method is not "automatic computation"; it depends on the existence of a reliable estimator of the standard deviation, $\hat{\sigma}(y_1, \ldots, y_n)$. In a lot of applications, this may not be available. Secondly, as pointed in [37.8], even with a reliable estimator of the standard deviation, the studentized bootstrap algorithm can be numerically unstable, resulting in very long confidence intervals. Third, the studentized bootstrap intervals are not "invariant under reparameterization".

BC_a Interval

The distribution of $\hat{\theta}^*$ is usually not symmetric but instead skewed to one side. The BC_a ("bias-corrected and accelerated") intervals were studied in [37.5–8] based on the bootstrap distribution. The BC_a intervals depend on two numerical parameters: a bias-correction parameter z_0 and an acceleration a. The upper endpoint $\hat{\theta}_{BC_a}[\alpha]$ of a one-sided level-α BC_a interval is defined as

$$\hat{\theta}_{BC_a}[\alpha] = \hat{G}^{-1}\left[\Psi\left(z_0 + \frac{z_0 + z^\alpha}{1 - a(z_0 + z^\alpha)}\right)\right] ,$$

where $\hat{G}$ is the empirical distribution function of the bootstrap samples, and Ψ is the standard normal cdf with $z^\alpha = \Psi^{-1}(\alpha)$.

The bias-correction parameter z_0 is usually estimated from the bootstrap sample as

$$\hat{z}_0 = \Psi^{-1}\left\{B^{-1}\sum_{b=1}^{B} I\left[\hat{\theta}^*(b) < \hat{\theta}\right]\right\} .$$

On the other hand, the acceleration parameter a is more subtle and cannot be estimated as directly from the bootstrap sample. *Di Ciccio* and *Efron* [37.8] presented several ways to estimate the acceleration parameter a. The second-order accuracy of the BC_a intervals is discussed in [37.8].

The BC_a intervals are "invariant under reparameterization". Under some conditions, the BC_a intervals are second-order-accurate. However, the BC_a intervals depend on the acceleration parameter a, which cannot be estimated directly from the bootstrap replications. The need to estimate a makes the BC_a method less intuitive to users. Therefore, the BC_a intervals are not obtained via "automatic computation".

ABC Method

The ABC method (short for "approximate bootstrap confidence" interval) is an analytic version of BC_a applied to smoothly defined parameters in exponential families. Instead of estimating z_0 using a bootstrap distribution as in the BC_a method, the ABC method estimates z_0 and the acceleration parameter a analytically. The ABC method requires one further estimate of a nonlinearity parameter. Based on these estimates, we can then construct ABC intervals. *DiCiccio* and *Efron* [37.8] provide the details of this ABC method, and they also show the second-order accuracy of this method. The ABC intervals are "invariant under reparameterization", but are not obtained via "automatic computation".

Prepivoting Method (Bootstrap Calibration)

Calibration is a bootstrap technique for getting confidence intervals accurate to higher orders. Concepts related to it have been proposed and studied in [37.9–12].

Suppose that $\hat{\theta}[\alpha]$ is the upper endpoint of a one-side level-α approximate confidence bound for parameter θ. Let

$$\gamma(\alpha) = P\left[\theta < \hat{\theta}(\alpha)\right]$$

be the calibration curve. If the approximation is perfect, then $\gamma(\alpha) = \alpha$ for any given α. Otherwise, we can use the calibration curve. For example, if $\gamma(0.03) = 0.025$ and $\gamma(0.96) = 0.975$, then we can use $\left(\hat{\theta}[0.03], \hat{\theta}[0.96]\right)$ instead of $\left(\hat{\theta}[0.025], \hat{\theta}[0.975]\right)$ as our approximate 0.95-level confidence interval.

In applications, we do not usually know the calibration curve $\gamma(\alpha)$. But we can use the bootstrap method to estimate $\gamma(\alpha)$ as follows:

$$\hat{\gamma}(\alpha) = P_*\left(\hat{\theta} < \hat{\theta}[\alpha]^*\right) ,$$

where P_* indicates the bootstrap sample and $\hat{\theta}[\alpha]^*$ is the upper α bound based on the bootstrap sample.

We can use bootstrap calibration asymptotically to obtain a higher order confidence interval from a given system of confidence intervals. Therefore, it can be applied to all of the methods reviewed in this chapter. For example, we can use bootstrap calibration to obtain third-order-accurate confidence intervals from studentized bootstrap intervals. However, bootstrap calibration involves more computation. For example, if we use $B = 1000$ (bootstrap sample size), then the bootstrap calibration will require 1 000 000 recomputations of the original statistic $\hat{\theta}$. In practice, the sample size n is usually not very large, so we can usually use one bootstrap calibration.

37.2.3 Linear Regression

The bootstraps discussed so far are based on iid samples, but in many applications this assumption does not hold. Consider the linear model

$$Y_i = \boldsymbol{x}_i\boldsymbol{\beta} + e_i ,$$

where $\boldsymbol{x}_i$ is a $k\times 1$ vector which may be a random or fixed variable. Here $\boldsymbol{\beta}$ is the $k\times 1$ parameter vector of interest, and $e_1, \ldots, e_n$ are uncorrelated errors with means of zero and variances of $\mathrm{Var}(e_i) = \sigma_i^2$, $i = 1, \ldots, n$, respectively. We assume that e_i and $\boldsymbol{x}_i$ are uncorrelated for all i when $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ are random. Let

$$\boldsymbol{Y} = (Y_1, \ldots, Y_n)^{\mathrm{T}} , \boldsymbol{e} = (e_1, \ldots, e_n)^{\mathrm{T}} ,$$
$$\text{and } \mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$$

The least square estimator is then

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{Y} .$$

Here $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is assumed to be nonsingular. Let $\boldsymbol{y} = (y_1, \ldots, y_n)$ denote the observed $\boldsymbol{Y}$.

When $e_1, \ldots, e_n$ are independent and identically distributed $\left(\sigma_i^2 = \sigma^2 \text{ for all } i\right)$, [37.2] proposed the following bootstrap method based on residuals. Let $r_i = y_i - \boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}$, $i = 1, \ldots, n$. We can treat $r_1, \ldots, r_n$ as observations of $e_1, \ldots, e_n$. We can resample $r_1^*, \ldots, r_n^*$ from $(r_1, \ldots, r_n)$ with replacement. Now define the bootstrap sample as

$$y_i^* = \boldsymbol{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + r_i^* , i = 1, \ldots, n .$$

Let $\boldsymbol{Y}^* = (y_1^*, \ldots, y_n^*)^{\mathrm{T}}$. The corresponding bootstrap estimator is

$$\hat{\boldsymbol{\beta}}^* = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{Y}^* .$$

Based on these bootstrap estimators, we can then apply the techniques in Sect. 37.2.1 and Sect. 37.2.2 to estimate the standard deviation of $\hat{\boldsymbol{\beta}}$ and the confidence intervals for $\boldsymbol{\beta}$. However, when $\boldsymbol{x}_i$ are random and the σ_i^2 values are not the same, Efron's bootstrap, which is based on resampling the residuals, does not provide a consistent result.

To deal with this heteroscedasticity, *Freedman* [37.13] suggests the following "Pair" bootstrap: resample $\left(\boldsymbol{x}_1^*, y_1^*\right), \ldots, \left(\boldsymbol{x}_n^*, y_n^*\right)$ from $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ with replacement and compute the bootstrap least squares estimate

$$\hat{\boldsymbol{\beta}}^* = \left(\mathbf{X}^{*\mathrm{T}}\mathbf{X}^*\right)^{-1}\mathbf{X}^{*\mathrm{T}}\boldsymbol{Y}^* ,$$

where $\mathbf{X}^* = \left(\boldsymbol{x}_1^*, \ldots \boldsymbol{x}_n^*\right)^{\mathrm{T}}$ and $\boldsymbol{Y}^* = \left(y_1^*, \ldots, y_n^*\right)^{\mathrm{T}}$. This method is consistent for heteroscedastic errors.

Hu and *Zidek* [37.14] propose another bootstrap method based on the observation that the estimator $\hat{\boldsymbol{\beta}}$ can be rewritten as

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{Y} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\sum_{i=1}^{n}\boldsymbol{x}_i y_i = \boldsymbol{\beta} + (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\sum_{i=1}^{n}\boldsymbol{x}_i e_i \,.$$

If we treat $z_i = \boldsymbol{x}_i r_i$ $(i = 1, \ldots, n)$ as an estimate of $\boldsymbol{x}_i e_i$, it is natural to suggest that the bootstrap estimator is:

$$\hat{\boldsymbol{\beta}}^* = \hat{\boldsymbol{\beta}} + \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\sum_{i=1}^{n} z_i^* \,,$$

where $z_1^*, \ldots, z_n^*$ is the bootstrap sample, which is drawn from $(z_1, \ldots, z_n)$ with replacement. This bootstrap method is also consistent for heteroscedastic errors.

As pointed out in [37.14], the numerical result of Freedman's "pair" bootstrap can be unstable. This is because the design matrix $\mathbf{X}^*$ changes for each bootstrap sample. For the bootstrap method proposed by *Hu* and *Zidek* [37.14], the design matrix $\mathbf{X}$ maintains the sample for each bootstrap sample. This is very important for cases with a small sample size n. When we applied the studentized bootstrap to both methods with heteroscedastic errors, *Hu* and *Zidek's* bootstrap was found to be easy to extend and has substantial numerical advantages over the "pair" bootstrap [37.14].

37.2.4 Some Remarks

We have discussed four second-order-accurate bootstrap methods. These methods are mainly useful for simple situations. However, resample methods are often needed in complex situations, such as nonlinear estimators and models with high-dimensional parameters. In these situations, there are clearly several difficulties that are encountered when using the traditional bootstraps: (i) it is difficult to derive an estimate for the acceleration parameter for the $\mathrm{BC_a}$ and ABC methods; (ii) for models with high-dimensional parameters, it is difficult to apply the studentized bootstrap and the prepivoting method; (iii) models with high-dimensional parameters are computationally intensive; (iv) the bootstrap sample may be quite different from the original sample which may produce unstable results.

In the following two sections, we will describe two recent proposals intended for complex models. The estimating function (EF) bootstrap is designed for estimates obtained from estimating equations. We show that the studentized estimating function bootstrap has the three desired properties. The Markov chain marginal bootstrap (MCMB) is mainly used to reduce computation in models with high-dimensional parameters.

37.3 Bootstrap Based on Estimating Equations

The traditional bootstrap methods based involve resampling the original data over and over again. Typically, the estimator is obtained from some estimating equation (*Godambe* and *Kale* [37.15]). The estimating function (EF) bootstrap proposed by [37.1, 14, 16] emphasizes the estimating function and the equation from which the estimator is obtained.

Following the same notations used by *Hu* and *Kalbfleisch* [37.1], let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be a sequence of independent random vectors of dimension q, and $\boldsymbol{\theta} \in \Omega \subset \mathbb{R}^p$ be an unknown parameter vector. For specified functions $\{\boldsymbol{g}_i\} : R^q \to R^p$, suppose that $E[\boldsymbol{g}_i(\boldsymbol{y}_i, \boldsymbol{\theta})] = 0$ for all $i = 1, \ldots, n$ and $\boldsymbol{\theta} \in \Omega$. We suppose that $\hat{\theta}$ is the solution of the following *unbiased linear estimating equation*

$$\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta}) = n^{-1/2}\sum \boldsymbol{g}_i(\boldsymbol{y}_i, \boldsymbol{\theta}) = 0 \,. \tag{37.1}$$

Here the normalizing constant $\left(n^{-1/2}\right)$ is chosen for the convenience of expressing asymptotic results. For simplicity, we also assume that $\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta})$ is a $1:1$ function of $\boldsymbol{\theta}$ and our main consideration will be the construction of confidence regions for the whole parameter vector $\boldsymbol{\theta}$, or for components or some functions of $\boldsymbol{\theta}$.

When the random vector $\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta})$ is exactly pivotal [37.17, 18], we can use exact methods to obtain confidence intervals or regions. However, in most cases, $\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta})$ is only approximately pivotal and we rely on asymptotic normality and χ^2 approximations to obtain the confidence intervals or regions of $\boldsymbol{\theta}$. Here we propose to use resampling methods to approximate the distribution of $\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta})$.

37.3.1 EF Bootstrap and Studentized EF Bootstrap

The EF Bootstrap

Let $\boldsymbol{z}_i = \boldsymbol{g}_i\left(\boldsymbol{y}_i, \hat{\theta}\right)$.

1. Draw a bootstrap sample $(z_1^*, \ldots, z_n^*)$ from $(z_1, \ldots, z_n)$ with replacement.
2. Compute $S^* = n^{-1/2} \sum z_i^*$. The bootstrap distribution of S^* can be used to approximate the distribution of $S(y, \theta)$.
3. Compute θ^* by solving $S(y, \theta) = S^*$.

The EF bootstrap generates a bootstrap sequence θ_j^* $(j = 1, \ldots, B$ where B is the bootstrap sample size) by repeating the above process B times. Based on θ_j^* $(j = 1, \ldots, B)$, we can then construct confidence regions of the parameter of interest (some functions of θ). Like Efron's bootstrap, this usually produces confidence intervals that are accurate only to the first order. *Hu* and *Kalbfleisch* [37.16] proposed one type of studentization. This studentization gives an approximation to second-order accuracy, but it is not invariant under reparameterization.

Here we introduce the studentized EF bootstrap proposed in [37.1]. We define

$$\mathbf{V}(y, \theta) = n^{-1} \sum [g_i(y_i, \theta) - \bar{g}][g_i(y_i, \theta) - \bar{g}]^{\mathrm{T}} , \tag{37.2}$$

where $\bar{g} = n^{-1} \sum g_i(y_i, \theta)$. In practice, we use the variance estimate

$$\hat{\mathbf{V}} = \mathbf{V}(y, \hat{\theta}) . \tag{37.3}$$

Instead of approximating the distribution of $S(y, \theta)$, we use a bootstrap method to approximate the distribution of

$$S_t(y, \theta) = \mathbf{V}(y, \theta)^{-1/2} S(y, \theta) . \tag{37.4}$$

In most cases, $S_t(y, \theta)$ is a better approximated pivotal.

Studentized EF Bootstrap

First we obtain $(z_1^*, \ldots, z_n^*)$ as in the EF bootstrap; compute

$$S_t^* = \mathbf{V}^{*-1/2} S^* ,$$

where $\mathbf{V}^* = n^{-1} \sum \left(z_i^* - \bar{z}^*\right)\left(z_i^* - \bar{z}^*\right)^{\mathrm{T}}$ and $\bar{z}^* = n^{-1} \sum z_i^*$, and finally solve

$$S_t(y, \theta) = S_t^* .$$

Under fairly general conditions [37.1], the studentized EF bootstrap is second-order-accurate and also invariant under reparameterization. The simplicity of its computation is discussed in the following two subsections.

37.3.2 The Case of a Single Parameter

When the parameter θ is a scalar and $S(y, \theta)$ is a monotonic function of θ, confidence intervals for θ based on the EF bootstrap are obtained as follows. For any specified α, we can find $S_{(\alpha)}^*$, the αth quantile of the bootstrap distribution of S^*. The two-sided interval $\left(\theta_{(\alpha/2)}^*, \theta_{(1-\alpha/2)}^*\right)$ obtained from

$$S\left(y, \theta_{(\alpha/2)}^*\right) = S_{(\alpha/2)}^*$$
$$\text{and} \quad S\left(y, \theta_{(1-\alpha/2)}^*\right) = S_{(1-\alpha/2)}^* ,$$

is the $100(1-\alpha)\%$ EF bootstrap confidence interval for θ. To obtain this interval, the equation $S(y, \theta) = S^*$ needs to be solved at only two points.

We can obtain higher order accuracy by using the studentized version based on (37.4). To do this, let $S_{t(\alpha)}^*$ be the αth quantile of the distribution of S_t^*. If $S_t(y, \theta)$ is monotonic in θ, then the equation $S_t(y, \theta) = S_{t(\alpha)}^*$ yields an endpoint for the interval. The confidence intervals obtained using the studentized EF bootstrap are usually second-order-accurate, and their performances are comparable to those of the $\mathrm{BC_a}$ and ABC methods [37.1].

From this simple model, we can see that the EF bootstrap has several advantages over Efron's bootstrap: (i) it is often computationally simpler, because we just have to solve the equation at two points; (ii) the studentized EF bootstrap is straightforward, while the classical studentized bootstrap requires a stable estimate of the variance; (iii) the studentized statistic $S_t(y, \theta)$ is invariant under reparameterization, as are the confidence intervals or regions based on studentized EF bootstrap. By contrast, the EF bootstrap is not invariant and it is usually first-order-accurate.

37.3.3 The Multiparameter Case

For a p-dimensional vector parameter θ, we use the approximate pivotal

$$\begin{aligned} Q(y, \theta) &= S(y, \theta)^{\mathrm{T}} \mathbf{V}(y, \theta)^{-1} S(y, \theta) \\ &= S_t(y, \theta)^{\mathrm{T}} S_t(y, \theta) . \end{aligned} \tag{37.5}$$

The distribution of $Q(y, \theta)$ can be approximated by the bootstrap distribution of

$$Q^* = S^{*\mathrm{T}} \mathbf{V}^{*-1} S^{*\mathrm{T}} = S_t^{*\mathrm{T}} S_t^*$$

using the calculations described in Sect. 37.3.1.

We define q^*_α to be the αth quantile of Q^*, which is determined by $P^*(Q^* > q^*_\alpha) = \alpha$. An approximate $100(1-\alpha)\%$ confidence region for $\boldsymbol{\theta}$ is then given by

$$C_{1-\alpha}(y) = \{\boldsymbol{\theta} : Q(y, \theta) \le q^*_\alpha\} . \tag{37.6}$$

This is based on the approximation

$$\begin{aligned} P[\boldsymbol{\theta} \in C_{1-\alpha}(y)] &= P[Q(\boldsymbol{y}, \boldsymbol{\theta}) \le q_\alpha] \\ &\approx P^*(Q^* \le q_\alpha) = 1-\alpha . \end{aligned} \tag{37.7}$$

Hu and *Kalbfleisch* [37.1] show that the confidence region in (37.7) is accurate up to order $O_p\left(n^{-3/2}\right)$. This improves on the usual χ^2 approximation, which is accurate up to order $O_p\left(n^{-1}\right)$. To construct the confidence region for a given confidence coefficient $1-\alpha$, one only needs to solve (37.6) for the relevant contour. This method is invariant under reparameterization.

The above approach does not generally work for inference on components or functions of $\boldsymbol{\theta}$. When $\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta})$ is a differentiable function of $\boldsymbol{\theta}$, *Hu* and *Kalbfleisch* [37.1] proposed a simple method based on some projections. However, the proposed method is usually accurate up to order $O_p\left(n^{-1}\right)$, and it is not invariant under reparameterization. When $\boldsymbol{S}(\boldsymbol{y}, \boldsymbol{\theta})$ is not differentiable, one needs to use the Markov chain marginal bootstrap (MCMB) proposed by *He* and *Hu* [37.19], which is introduced in Sect. 37.4.

37.3.4 Some Examples

Example 1. Estimating the population mean. Observations $y_1, \ldots, y_n$ are made on independent and identically distributed random variables, each with an unspecified distribution function, F. Interest focuses on the mean, μ, of F which is estimated with $\hat{\mu} = \bar{y}$. In the usual classical bootstrap (Efron's bootstrap), we (i) draw the bootstrap sample $\{y^*_1, \ldots, y^*_n\}$ from $\{y_1, \ldots, y_n\}$ and (ii) calculate the bootstrap sample mean $\hat{\mu}^*_C = n^{-1} \sum y^*_i$. These steps are repeated and the empirical distribution of the $\left(\hat{\mu}^*_C - \hat{\mu}\right)$ is the bootstrap approximation to the sampling distribution of $\hat{\mu} - \mu$.

In contrast, the EF bootstrap begins with the estimating equation $\sum(y_i - \mu) = 0$, whose solution is $\hat{\mu} = \bar{y}$. The component functions $y_i - \mu$ are estimated with $z_i = y_i - \bar{y}$, $i = 1, \ldots, n$. The method proceeds as follows: (i) draw a bootstrap sample $\{z^*_1, \ldots, z^*_n\}$ from $\{z_1, \ldots, z_n\}$; (ii) calculate $S^* = n^{-1/2} \sum z^*_i$. The bootstrap distribution of S^* approximates the sampling distribution of $S(\boldsymbol{y}, \mu) = \sqrt{n}(\hat{\mu} - \mu)$. Note that if μ^* is the solution to $S(\boldsymbol{y}, \mu) = S^*$, the bootstrap distribution of $\mu^* - \hat{\mu}$ approximates the distribution of $\mu - \hat{\mu}$.

The difference between the methods is evident, even though they give, in the end, identical results. With the classical bootstrap, $\hat{\mu}^*_C - \hat{\mu}$ approximates $\hat{\mu} - \mu$, whereas in the EF procedure, $\mu^* - \hat{\mu}$ approximates $\mu - \hat{\mu}$. As a consequence, μ^* is "bias corrected". The comparison between the studentized versions is similar.

Example 2. Common mean with known and unknown variances. Suppose that $y_1, \ldots, y_n$ are from populations with $Ey_i = \mu$ and $var(y_i) = \sigma^2_i$. When σ^2_i are known, the estimating equation,

$$\sum \frac{y_i - \mu}{\sigma^2_i} = 0$$

gives rise to the weighted least squares estimator,

$$\hat{\mu} = \left(\sum y_i/\sigma^2_i\right) / \left(\sum 1/\sigma^2_i\right) .$$

The EF and classical bootstraps can be applied to this problem in a straightforward way. (As noted above, the classical bootstrap is equivalent to the classical procedure of resampling $(y_i, \sigma_i), i = 1, \ldots, n$.) *Hu* and *Kalbfleisch* [37.16] compare the EF bootstrap with the classical bootstrap and the asymptotic normal approximation assuming normal and uniform errors. All methods do reasonably well, though the studentized versions of the EF and classical bootstraps do somewhat better than the other methods with abnormal errors.

Suppose there are k independent strata and in the ith stratum $y_{ij} \sim N\left(\mu, \sigma^2_i\right)$, $j = 1, \ldots, n_i$ independently, where $n_i \ge 3$ and $i = 1, \ldots, k$. The variances σ^2_i are unknown and interest centers on the estimation of μ. This problem has received much attention in the literature [37.20–24] [Bartlett (1936), Neyman and Scott (1948), Kalbfleisch and Sprott (1970), Barndorff-Nielsen (1983). *Neyman* and *Scott* (1948) showed that the maximum likelihood estimator can be inefficient. They (and many others) proposed the estimating equation

$$\sum_{i=1}^{k} \frac{n_i(n_i - 2)(\bar{y}_i - \mu)}{T_i(\mu)} = 0 ,$$

where $T_i(\mu) = \sum_{j=1}^{n_i}(y_{ij} - \mu)^2$ and $\bar{y}_i = \sum_{j=1}^{n_i} y_{ij}/n_i$. More generally, we could relax the condition of normal errors and still use the above equation to estimate μ.

When the number of strata k is large and the individual n_i's are small, usual inferential techniques can cause substantial difficulty. This is the case considered here, although other situations are also of interest and will be discussed elsewhere.

Let $y_i = (y_{i1}, \ldots, y_{in_i})$ and $g_i(y_i, \mu) = n_i(n_i - 2)(\bar{y}_i - \mu)/T_i(\mu)$. The estimating equation can therefore be rewritten as

$$\sum_{i=1}^{k} g_i(y_i, \mu) = 0 ,$$

and the EF bootstrap can now be applied in a straightforward manner.

37.4 Markov Chain Marginal Bootstrap

For statistical models with high-dimensional parameters, it is usually difficult to apply the bootstrap method because it is computationally intensive. For example, if one needs one minute to obtain the estimator, then one needs 1000 min to apply the bootstrap method to a sample of size $B = 1000$. To reduce the computational complexity of applying common bootstrap methods to high-dimensional parameters, *He* and *Hu* [37.19] propose the Markov chain marginal bootstrap (MCMB). In this section, we only review the MCMB for M-estimators of a linear model. Please see [37.19] for more general models and estimators.

Consider the linear regression problem $Y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + e_i$, $(i = 1, \cdots, n)$ with independently and identically distributed errors e_i. An M-estimator, $\hat{\boldsymbol{\beta}}$, solves

$$n^{-1} \sum_{i=1}^{n} \psi\left(Y_i - \boldsymbol{x}_i'\boldsymbol{\beta}\right) \boldsymbol{x}_i = 0 \qquad (37.8)$$

for a score function ψ. In most applications, the function ψ is bounded and continuous. An important exception is the least absolute deviation estimator with $\psi(r) = \mathrm{sgn}(r)$. In this case, the equation (37.8) may not be solved exactly, but minimizing $\sum_{i=1}^{n} |Y_i - \boldsymbol{x}_i'\boldsymbol{\beta}|$ over $\boldsymbol{\beta} \in \mathbb{R}^p$ guarantees a solution so that (37.8) holds approximately.

Under some suitable conditions [37.25], the estimator $\hat{\boldsymbol{\beta}}$ is consistent and asymptotically normal,

$$n^{1/2}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) \to N\left(0, \left\{E\psi^2(e)/[E\psi'(e)]^2\right\}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right) ,$$

where $\mathbf{X}$ is the design matrix. A direct estimate of the variance does not always produce reliable confidence levels for inference. This is because it is difficult to estimate the constant $\left[E\psi'(e)\right]^2$ in a lot of cases.

For example, consider the minimum L_d-norm estimator that minimizes $\sum_{i=1}^{n} |y_i - \boldsymbol{x}_i'\boldsymbol{\beta}|^d$ $(d = 1.5)$. In this case, $E\psi'(e) = 0.5E|e|^{-0.5}$. One needs to estimate the constant $E|e|^{-0.5}$ to construct a confidence interval based on the asymptotic variance. A natural estimator is the average of n absolute residuals $r_i = y_i - \boldsymbol{x}_i'\hat{\beta}_n$. When $n = 20$ and e has a standard normal distribution and the residuals resemble a random sample drawn from it, then a simple simulation shows that the average of $|r_i|^{-0.5}$ has a mean of 1.71 and standard error of 0.80. When one or a few residuals are very close to 0, the estimate could be very large. Therefore, the confidence intervals constructed from this estimated asymptotic variance would be poor.

To avoid estimating the asymptotic variance directly, one can use the usual bootstrap methods (residual bootstrap or pair-wise bootstrap). In this case, a p-dimensional nonlinear system has to be solved for each bootstrap sample. This can become a computational burden for large p. Also, the pair-wise bootstrap can be numerical unstable, because the design matrix changes for each bootstrap sample. The EF bootstrap or studentized EF bootstrap is often more stable because it uses all of the design points in each resample, but its computational complexity is no less than that of the usual bootstrap methods. When ψ is differentiable, one can solve the computational problem using projection [37.1]. However, ψ is not differentiable in a lot of cases.

MCMB overcomes the computational complexity by breaking up the p-dimensional system into p marginal (one-dimensional) equations. The algorithm proceeds as follows. Let subscript β_j be the jth component of $\boldsymbol{\beta}$ and subscript $\boldsymbol{\beta}^{(k)}$ be the kth iteration of the algorithm. Suppose that $\hat{\boldsymbol{\beta}}$ is the estimate from (37.8) and $r_i = y_i - \boldsymbol{x}_i\hat{\boldsymbol{\beta}}$ are the residuals. Let $z_i = \psi(r_i)\boldsymbol{x}_i$ be the scores. The jth component of z_i will be denoted by z_{ij} $(i = 1, \ldots, n$ and $j = 1, \ldots, p)$. For the kth iteration with $k = 0, 1, \ldots$, we perform

1. For the jth component, we resample $\left\{z_{ij}^*, i = 1, \ldots, n\right\}$ from $\left\{z_{ij}, i = 1, \ldots, n\right\}$ without replacement.

2. Let $s_j^{(k)} = \sum_{i=1}^n z_{ij}^*$ and solve $\beta_j^{(k)}$ from

$$\sum_i^n \psi(y_i - \sum_{l=1}^{j-1} x_{il}\beta_l^{(k)} - x_{ij}\beta_j - \sum_{l=j+1}^{n} x_{il}\beta_l^{(k-1)})x_{ij} = s_j^{(k)} . \quad (37.9)$$

These two steps are performed for $j = 1, \ldots, p$.

This algorithm yields a sequence $\boldsymbol{\beta}^{(0)} = \hat{\boldsymbol{\beta}}$, $\boldsymbol{\beta}^{(1)}, \ldots \boldsymbol{\beta}^{(k)}, \ldots$. It is clearly a Markov chain. This method is called the Markov chain marginal bootstrap (MCMB), since a resampling process (bootstrap) is used with each marginal equation (37.9). In fact, the MCMB shares two properties with MCMC. That is, both methodologies aim to break up a high-dimensional problem into several one-dimensional ones, and both yield Markov chains as products. However, we must note that MCMB does not use any MCMC algorithm, and it is not derived from the MCMC framework.

Now we explain why the MCMB method reduces the computational complexity of the usual bootstrap method. To generate an additional variate $\beta_j^{(k)}$, one needs to resample and solve a one-dimensional equation, both of which are of the complexity $O(n)$. For $\boldsymbol{\beta}^{(k)}$, the complexity is $O(np)$ for large n and p. However, common bootstrap methods have to solve a p-dimensional system. Even the simplest system (a linear system) requires of the order of $O(np^2)$ computations. Therefore, the MCMB method reduces the computational complexity for large p. Some other studies have been discussed in [37.19].

Like the EF bootstrap, the MCMB method has another advantage; that all of the design points are used in each iteration. This leads to more reliable numerical results, especially when there are leverage points present in the data, as compared to the pairwise bootstrap method that can suffer from poor bootstrap estimates when a leverage point is excluded or duplicated in a resample.

The MCMB method can be used for the maximum likelihood estimators from general parametric models. The asymptotic validity of the MCMB method for general parametric models has been given in [37.19]. The use of MCMB for general M-estimators (or GEE estimators) is explored in [37.26].

The MCMB is usually not invariant under reparameterization. *He* and *Hu* [37.19] also show that the MCMB is first-order-accurate. However, it is unknown whether MCMB is second-order-accurate. Future research is clearly needed to understand the MCMB method.

37.5 Applications

In this section, we will apply the above bootstraps to two examples. The first example is a simple linear model. We use this example to illustrate the MCMB algorithm and show why the MCMB bootstrap works. The second example involves a linear estimating equation from L_q estimation. For more discussions of these examples, please refer to [37.1, 19].

Example 1. Simple linear model. First, we consider a simple regression model with sample size n and $p = 2$. In this special case, we have

$$n^{-1/2}\sum_{i=1}^n \left(Y_i - \boldsymbol{x}_{i1}\hat{\beta}_1^{(k)} - \boldsymbol{x}_{i2}\hat{\beta}_2^{(k-1)}\right)\boldsymbol{x}_{i1} = d_1^{(k)}$$

and

$$n^{-1/2}\sum_{i=1}^n \left(Y_i - \boldsymbol{x}_{i1}\hat{\beta}_1^{(k)} - \boldsymbol{x}_{i2}\hat{\beta}_2^{(k)}\right)\boldsymbol{x}_{i2} = d_2^{(k)} ,$$

where $d_1^{(k)} = n^{-1/2}\sum_{i=1}^n x_{i1}e_{i1}^{*(k)}$ and $d_2^{(k)} = n^{-1/2} \times \sum_{i=1}^n \boldsymbol{x}_{i2}e_{i2}^{*(k)}$, and both $e_{i1}^{*(k)}$ and $e_{i2}^{*(k)}$ are drawn independently with replacement from $r_i = Y_i - \hat{\beta}_1\boldsymbol{x}_{i1} - \hat{\beta}_2\boldsymbol{x}_{i2}$ $(i = 1, \cdots, n)$, the residuals from the parameter estimate $(\hat{\beta}_1, \hat{\beta}_2)$. Now let $s_{11} = n^{-1}\sum_{i=1}^n \boldsymbol{x}_{i1}^2$, $s_{12} = n^{-1}\sum_{i=1}^n \boldsymbol{x}_{i1}\boldsymbol{x}_{i2}$ and $s_{22} = n^{-1}\sum_{i=1}^n \boldsymbol{x}_{i2}^2$. Then the two equations can be written as

$$s_{11}n^{1/2}\left(\hat{\beta}_1 - \hat{\beta}_1^{(k)}\right) = d_1^{(k)} - s_{12}n^{1/2}\left(\hat{\beta}_2 - \hat{\beta}_2^{(k-1)}\right) ,$$

$$s_{22}n^{1/2}\left(\hat{\beta}_2 - \hat{\beta}_2^{(k)}\right) = d_2^{(k)} - s_{12}n^{1/2}\left(\hat{\beta}_1 - \hat{\beta}_1^{(k)}\right) .$$

Note that the right hand sides of the above equations are sums of two independent variables, so by using variance–covariance operation and assuming that the covariance matrix of $n^{1/2}\left(\hat{\beta} - \hat{\beta}^{(k)}\right)$ stabilizes to $V = (v_{ij})_{2\times 2}$ as $k \to \infty$, we have

$$s_{11}^2 v_{11} = s_{11}\sigma^2 + s_{12}^2 v_{22} ,$$
$$s_{22}^2 v_{22} = s_{22}\sigma^2 + s_{12}^2 v_{11} ,$$
$$s_{22}v_{12} = -s_{12}v_{11} .$$

Using some simple calculations, we can show that $\mathbf{V} = \sigma^2[(s_{ij})_{2\times 2}]^{-1}$. That is, the bootstrap variance–

covariance of $n^{1/2}\left(\hat{\boldsymbol{\beta}}-\hat{\boldsymbol{\beta}}^{(k)}\right)$ stabilizes to the desired asymptotic covariance matrix for the least squares estimator.

Now we move to a real example about grade point prediction. The director of admissions of a small college administered a newly designed entrance test to eight students selected at random from the new freshman class in a study to determine whether a student's grade point average (GPA) at the end of the freshman year (Y) can be predicted from their entrance test score (x).

The eight pairs of scores were: (5.5, 3.1), (4.8, 2.3), (4.7, 3.0), (5.9, 3.8), (4.1, 2.2), (4.7, 1.5), (4.5, 3.0) and (5.3, 3.6). After we fit the linear regression, we get the estimated regression line

$$\hat{Y} = -1.646 + 0.903x\,.$$

The residuals are $r = (-0.22, -0.39, 0.53, 0.12, 0.14, -1.10, 0.57, 0.45)$. For the above MCMB algorithm, we have $n = 8$, $x_{i1} = 1$ for $i = 1, \ldots, 8$, $(x_{12}, \ldots, x_{82}) = (5.5, \ldots, 5.3)$. Then we can apply the MCMB algorithm to $k = 200$ to get the 95% confidence intervals: β_1: $[-5.526, 2.134]$ and β_2: [0.138,1.668]. In this example, it is very easy to calculate the confidence intervals from other methods, but we are just using it to show how MCMB can be applied. In this simple example, there is no advantage to using the MCMB method. As we mentioned earlier, the main advantage of the MCMB is that it works well for the following two cases: (i) high-dimensional parameters, and (ii) estimating equations that are not differentiable. More complete simulation studies can be found in [37.19].

Example 2. The L_q estimation. Consider a linear estimating equation in which $g_i(y_i, \theta)$ is not differentiable with respect to θ. Such situations are quite common in nonparametric and semiparametric models [37.27, 28] and in robust regression [37.25]. Estimating functions that are not differentiable can give rise to various difficulties. Classical statistical results do not apply in general, and other methods (bootstrap methods) for confidence interval estimation are needed.

We consider the general regression model

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + e_i, i = 1, \ldots, n\,, \quad (37.10)$$

and suppose that β is to be estimated by minimizing

$$\sum_{i=1}^{n} |y_i - (\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i})|^{1.5}\,.$$

The corresponding estimating equation

$$\sum_{i=1}^{n} \operatorname{sgn}\left(y_i - \boldsymbol{x}_i^{\mathrm{T}}\beta\right)\, \boldsymbol{x}_i\, |y_i - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}|^{1/2} = 0\,, \quad (37.11)$$

where $\boldsymbol{x}_i = (1, x_{1i}, x_{2i})^{\mathrm{T}}$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^{\mathrm{T}}$. The EF bootstrap procedure for estimating the whole parameter $\boldsymbol{\beta}$ or components of $\boldsymbol{\beta}$ can be applied to this problem in a straightforward manner.

Consider a fixed design where $n = 20$, $\boldsymbol{x}_1 = (1.27, -1.10, 2.19, 0.73, -0.07, 0.42, 0.37, 0.45, -0.78, 0.76, 0.44, 1.32, -0.40, 0.33, -0.40, 0.55, 0.51, -0.11, -1.15, 1.71)$, and $\boldsymbol{x}_2 = (1.60, 1.09, -0.02, -0.83, 3.05, 0.34, -0.87, 0.45, -0.78, 0.76, 0.44, 1.32, -0.40, 0.33, -1.85, 0.69, 0.11, 1.47, 0.87, 0.12)$, and y_i are generated from (37.10) with $\beta_0 = \beta_1 = \beta_2 = 1$.

For the whole parameter vector $\boldsymbol{\beta}$, the studentized estimating function bootstrap method can be used to obtain a highly accurate $O_p\left(n^{-3/2}\right)$ confidence region. Here we just report a result based on 1000 simulations. For each simulation, we can construct a 95% confidence region for $\boldsymbol{\beta}$. Of the 1000 confidence regions, 963 confidence regions cover the true parameter $\boldsymbol{\beta} = (1, 1, 1)$.

For single parameters, the estimating function bootstrap method depends on whether the estimating function is differentiable. In this example, the estimating function is not differentiable at the point 0, so we cannot use the simple method proposed by *Hu* and *Kalbfleisch* [37.1]. In this case, the MCMB method can be used to construct the confidence interval for each component of $\boldsymbol{\beta}$.

The average confidence interval in Table 37.1 is obtained by taking the averages of the two end points of the intervals over 500 cases. We consider three methods here. NORM represents the usual confidence interval

Table 37.1 Minimum L_q distance estimator ($q = 1.5$). Simulated coverage probabilities and average confidence intervals (fixed design)

	β_0		β_1		β_2	
MCMB	90.8	[0.56, 1.43]	89.0	[0.55, 1.46]	87.6	[0.66, 1.33]
NORM	76.8	[0.63, 1.35]	76.4	[0.62, 1.38]	75.6	[0.72, 1.29]
PAIR	88.0	[0.53, 1.43]	86.0	[0.52, 1.50]	86.2	[0.62, 1.38]

based on normal approximation. PAIR represents the paired bootstrap introduced in Sect. 37.2.

For each of the 500 samples there is an estimate of β_0, β_1 and β_2. Based on these estimators, we can calculate the standard deviations, and they are 0.24, 0.28 and 0.22 respectively. The confidence intervals, constructed from these estimators using the standard formula of the average plus or minus 1.64 times the SD, are [0.56,1.42], [0.55,1.48] and [0.66,1.34], respectively. We may use these three intervals as benchmarks for the other methods under consideration. It is clear from Table 37.1 that MCMB performed well.

37.6 Discussion

We have reviewed different bootstrap methods for independent observations. However, for a lot of applications, the observations may depend on each other. For stationary processes, several bootstrap procedures have been proposed, which include the block bootstrap and others. The estimating function bootstrap can also be extended to dependent observations. *Hu* and *Kalbfleisch* [37.29] considered linear and nonlinear autoregressive models.

One important application of bootstrap is in longitudinal data analysis. In this application, a generalized estimating equation (GEE) is usually available. Within each stratum (for each patient), the observations are dependent. But the observations are independent between stratums. The estimating function bootstrap can be applied as in the common mean problem in Sect. 37.3. However, some modifications are necessary to apply the classical bootstrap procedures.

Major problems with using bootstrap for high-dimensional parameters include that it is computationally intensive and can produce unreliable numerical results. The estimating function bootstrap method solves this problem by fixing one side of the estimating equation. When the estimating function is differentiable, we can use EF bootstrap to construct confidence intervals (regions). When the estimating function is not differentiable, we can then use the MCMB to solve this problem by considering a one-dimensional equation at each step.

In applications, it is also important to choose the bootstrap sample size B appropriately. When the original process (to get the estimator, $\hat{\theta}$) does not involve intensive computation, $B = 1000$ or 2000 is recommended. In general, to estimate the variance–covariance matrix, we may only need a bootstrap sample size of 100 to 200. For confidence intervals based on quantiles, it would be better to use $B = 1000$ or 2000.

References

37.1 F. Hu, J. D. Kalbfleisch: The estimating equation bootstrap (with discussions), Can. J. Stat. **28**, 449–499 (2000)

37.2 B. Efron: Bootstrap methods: another look at the jackknife, Ann. Stat. **7**, 1–26 (1979)

37.3 B. Efron: *The jackknife, the bootstrap, and other resampling plans*, Soc. Ind. Appl. Math. CBMS-Natl. Sci. Found. Monogr., Vol. 38 (SIAM, Philadelphia 1982)

37.4 P. Hall: *The Bootstrap and Edgeworth Expansion*. (Springer, Berlin Heidelberg 1992)

37.5 B. Efron: Better bootstrap confidence intervals (with discussion), J. Am. Stat. Assoc. **82**, 171–200 (1987)

37.6 B. Efron, R. J. Tibshirani: Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, Stat. Sci. **1**, 54–77 (1986)

37.7 T. J. Diciccio, J. P. Romano: A review of bootstrap confidence intervals, J. Roy. Stat. Soc. **50**, 338–354 (1988)

37.8 T. J. Diciccio, B. Efron: Bootstrap confidence intervals (with discussion), Stat. Sci. **11**, 189–228 (1996)

37.9 P. Hall: Theoretical comparison of bootstrap confidence intervals (with discussion), Ann. Stat. **16**, 927–985 (1988)

37.10 W. Y. Loh: Calibrating confidence coefficients, J. Am. Stat. Assoc. **82**, 155–162 (1987)

37.11 R. Beran: Prepivoting to reduce level error of confidence sets, Biometrika **74**, 457–468 (1987)

37.12 P. Hall, M. A. Martin: Comment on paper by DiCiccio and Efron, Stat. Sci. **11**, 189–228 (1996)

37.13 D. A. Freedman: Bootstrapping regression models, Ann. Stat. **9**, 1218–1228 (1981)

37.14 F. Hu, J. V. Zidek: A bootstrap based on the estimating equations of the linear model, Biometrika **82**, 263–275 (1995)

37.15 V. P. Godambe, B. K. Kale: *Estimating Functions*, ed. by V. P. Godambe (Oxford Univ. Press, Oxford 1992) pp. 3–20

37.16 F. Hu, J. D. Kalbfleisch: Estimating equations and the bootstrap. In: *Selected Proceedings of the Symposium on Estimating Equations.*, IMS Lect. Note Monogr. Ser., Vol. 32, ed. by I. V. Basawa, V. P. Godambe, R. L. Taylor (Institute of Mathematical Statistics, Hayward 1997) pp. 405–416
37.17 R. J. Buehler: Fiducial inference. In: *Encyclopedia of Statistical Sciences*, Vol. 3, ed. by Wiley (S. Kotz and N. L. Johnson, New York 1983) pp. 76–79
37.18 M. I. Parzen, L. J. Wei, Z. Ying: A resampling method based on pivotal estimating functions, Biometrika **81**, 341–50 (1994)
37.19 X. He, F. Hu: Markov chain marginal bootstrap, J. Am. Stat. Assoc. **97**, 783–795 (2002)
37.20 D. R. Cox, N. M. Reid: Parameter orthogonality and approximate conditional inference (with discussion), J. R. Stat. Soc. **49**, 1–39 (1987)
37.21 M. S. Bartlett: The information available in small samples, Proc. Camb. Phil. Soc. **34**, 33–40 (1936)
37.22 O. E. Barndorff-Nielsen: On a formula for the distribution of a maximum likelihood estimator, Biometrika **70**, 343–365 (1983)
37.23 J. D. Kalbfleisch, D. A. Sprott: Application of likelihood methods to models involving large numbers of nuisance parameters (with discussion), J. R. Stat. Soc. B **32**, 175–208 (1970)
37.24 Neyman, Scott: Consistent estimates based on partially consistent observations, Econometrica **16**, 1–32 (1948)
37.25 P. J. Huber: *Robust Statistics*. (Wiley, New York 1981)
37.26 M. Kocherginsky: Contributions to Bootstrap-Based Inference in Linear and Nonlinear Models, Ph.D. Dissertation (2002)
37.27 R. Koenker, G. J. Bassett: Regression quantiles, Econometrica **84**, 33–50 (1978)
37.28 T. P. Hettmansperger: *Statistical Inference Based on Ranks*. (Wiley, New York 1984)
37.29 F. Hu and J. D. Kalbfleisch: *An estimating function bootstrap for linear and non-linear autoregressive models*, unpublished

38. Random Effects

This chapter includes well-known as well as state-of-the-art statistical modeling techniques for drawing inference on correlated data, which occur in a wide variety of studies (during quality control studies of similar products made on different assembly lines, community-based studies on cancer prevention, and familial research of linkage analysis, to name a few).

The first section briefly introduces statistical models that incorporate random effect terms, which are increasingly being applied to the analysis of correlated data. An effect is classified as a random effect when inferences are to be made on an entire population, and the levels of that effect represent only a sample from that population.

The second section introduces the linear mixed model for clustered data, which explicitly models complex covariance structure among observations by adding random terms into the linear predictor part of the linear regression model. The third section discusses its extension – generalized linear mixed models (GLMMs) – for correlated nonnormal data.

The fourth section reviews several common estimating techniques for GLMMs, including the EM and penalized quasi-likelihood approaches, Markov chain Newton-Raphson, the stochastic approximation, and the S-U algorithm. The fifth section focuses on some special topics related to hypothesis tests of random effects, including score tests for various models. The last section is a general discussion of the content of the chapter and some other topics relevant to random effects models.

38.1 Overview

Classical linear regression models are a powerful tool for exploring the dependence of a response (such as blood pressure) on explanatory factors (such as weight, height and nutrient intake). However, the normality assumption required for these response variables has severely limited its applicability. To accommodate a wide variety of independent nonnormal data, *Nelder* and *Wedderburn* [38.1] and *McCullagh* and *Nelder* [38.2] introduced generalized linear models (GLMs), a natural generalization of linear regression models. The GLMs allow responses to have nonGaussian distributions. Hence, data on counts and proportions can be conveniently fitted into this framework. In a GLM, the mean of a response is typically linked to linear predictors via a nonrandom function, termed the *link function*. For analytical convenience, the link function is often determined by

the response's distribution. As an example, for Poisson data, the link is routinely chosen as log, whereas for Bernoulli responses, the link is usually chosen to be logit.

In many applications, however, responses are correlated due to unobservable factors, such as circumstantial or genetic factors. Consider the problem of investigating the strength of the beams made by randomly selected manufacturers. Beams made at the same factory are likely to be correlated because the were made using the same manufacturing procedures. Other examples include a longitudinal study of blood pressure, where repeated observations taken from the same individuals are likely be correlated, and a familial study in cardiovascular disease, where the incidents of heart failure from family members are likely to be related. In the last two decades, random effects models have emerged as a major tool for analyzing these kinds of correlated data (see [38.3–7] among others).

Indeed, using random effects in the modeling of correlated data provides several benefits. First, it provides a framework for performing data modeling in unbalanced designs, especially when measurements are made at arbitrary irregularly spaced intervals over many observational studies (as opposed to ANOVA, which requires a balanced dataset). Secondly, random effects can be used to model subject-specific effects, and they offer a neat way to separately model within- and between-subject variations. Thirdly, the framework of random effects provides a systematic way to estimate or predict individual effects.

Though conceptually attractive, GLMMs are often difficult to fit because of the intractability of the underlying likelihood functions. Only under special circumstances, such as when both response and random effects are normally or conjugately distributed, will the associated likelihood function have a closed form. Cumbersome numerical integrations often have to be performed. To alleviate this computational burden, various modeling techniques have been proposed. For example, *Stiratelli* et al. [38.4] proposed an EM algorithm for fitting serial binary data; *Schall* [38.5] developed an iterative Newton-Raphson algorithm; *Zeger* and *Karim* [38.6] and *McCulloch* [38.7] considered Monte Carlo EM methods. All of these commonly used inferential procedures will be presented and discussed in this chapter.

The rest of this chapter is structured as follows. Section 38.2 introduces the linear mixed model for clustered data and Sect. 38.3 discusses its extension, generalized linear mixed models, for correlated nonnormal data. Section 38.4 reviews several common estimating techniques for GLMMs, including the EM approach, penalized quasi-likelihood, Markov chain Newton-Raphson, the stochastic approximation and the S–U algorithm. Section 38.5 focuses on some special topics related to hypothesis tests of random effects. Section 38.6 concludes this chapter with discussion and some other topics relevant to random effects models.

Throughout this chapter, $f(\cdot)$ and $F(\cdot)$ denote the probability density (or probability mass) function (with respect to some dominating measure, such as the Lebesgue measure) and the cumulative distribution function, respectively. If the context is clear, we do not use separate notations for random variables and their actual values.

38.2 Linear Mixed Models

A clustered data structure is typically characterized by a series of observations on each of a collection of observational clusters. Consider the problem of investigating whether the beam produced from iron is more resilient than that from an alloy. To do this, we measure the strength of the beams made of iron and alloy from randomly selected manufacturers. Each manufacturer may contribute multiple beams, in which case each manufacturer is deemed as a cluster, while each beam contributes to a unit of observation. Other examples include the measurements of products produced by a series of assembly lines, and blood pressure taken weekly on a group of patients, in which cases the clusters are assembly lines and patients respectively. Clustering typically induces dependence among observations. A linear mixed model [38.3] explicitly models the complex covariance structure among observations by adding random terms into the linear predictor part of a linear regression model. Thus, both random and fixed effects will be present in an LMM. In data analysis, the decision on whether a factor should be fixed or random is often made on the basis of which effects vary with clusters. That is, clusters are deemed to be a random sample of a larger population, and therefore any effects that are not constant for all clusters are regarded as random.

As an example, let's say that $\boldsymbol{Y}_i$ denotes the response vector for the ith of a total of m clusters, where n_i measurements of blood pressure were taken for the

ith patient. $\mathbf{X}_i$ the known covariate matrix ($n_i \times p$) associated with the observations, such as the patient's treatment assignment and the time when the observation was taken, $\boldsymbol{b}_i$ is the vector of random effects and $\mathbf{Z}_i$ is the known design matrix associated with the random effects. Usually, the columns of $\mathbf{Z}_i$ are vectors of ones and a subset of those of $\mathbf{X}_i$ for modeling random intercepts and slopes. A linear mixed model can thus be specified as

$$\boldsymbol{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i \,, \tag{38.1}$$

where we typically assume that the random error vector $\boldsymbol{\epsilon}_i \sim MVN(0, \sigma^2\mathbf{I}_{n_i})$ and $\boldsymbol{\epsilon}_i$ is independent of $\boldsymbol{b}_i$, which is assumed to have an expectation of zero for model identifiability. Here, $\mathbf{I}_{n_i}$ is an identity matrix of order n_i. In practice, we often assume $\boldsymbol{b}_i \sim MVN(0, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$, where its variance–covariance matrix is dependent on a fixed q-dimensional (a finite number) parameter, say, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)'$, termed "variance components". These variance components convey information about the population that the clusters are randomly selected from and are often of interest to practitioners, aside from the fixed effects.

To encompass all data, we denote the concatenated collections of $\boldsymbol{Y}_i$'s, $\mathbf{X}_i$'s, $\boldsymbol{b}_i$'s and $\boldsymbol{\epsilon}_i$'s by $\boldsymbol{Y}, \mathbf{X}, \boldsymbol{b}, \boldsymbol{\epsilon}$. For example, $\boldsymbol{Y} = (\mathbf{Y}_1', \ldots, \mathbf{Y}_m')'$. We now denote a block diagonal matrix whose ith diagonal block is $\mathbf{Z}_i$ by $\mathbf{Z}$. In this case (38.1) can be rewritten compactly as

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{b} + \boldsymbol{\epsilon} \,, \tag{38.2}$$

where $\boldsymbol{b} \sim MVN(0, \mathbf{D})$, $\boldsymbol{\epsilon} \sim MVN(0, \sigma^2\mathbf{I}_{\mathcal{N}})$ and $\boldsymbol{b}$ and $\boldsymbol{\epsilon}$ are independent. Here, $\mathbf{D}$ is a block diagonal matrix whose diagonal blocks are $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, and $\mathbf{I}_{\mathcal{N}}$ is an identity matrix of order $\mathcal{N}$, where $\mathcal{N}$ is the total number of observations (so $\mathcal{N} = \sum_{i=1}^{m} n_i$).

Indeed, model (38.2) accommodates a much more general data structure beyond clustered data. For example, with properly defined $\mathbf{Z}$ and random effects $\boldsymbol{b}$, model (38.2) encompasses crossed factor data [38.8] and Gaussian spatial data [38.9].

38.2.1 Estimation

Fitting model (38.1) or its generalized version (38.2) is customarily likelihood-based. A typical maximum likelihood estimation procedure is as follows.

First observe that $\boldsymbol{Y}$ is normally distributed, $\boldsymbol{Y} \sim MVN(\mathbf{X}\boldsymbol{\beta}, \mathbf{V})$, where $\mathbf{V} = \mathbf{Z}\mathbf{D}\mathbf{Z}' + \sigma^2\mathbf{I}_{\mathcal{N}}$, so that the log-likelihood for the observed data is

$$\ell = -\frac{1}{2}(\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}) - \frac{1}{2}\log|\mathbf{V}| - \frac{\mathcal{N}}{2}\log 2\pi \,. \tag{38.3}$$

Denote the collection of unknown parameters in the model by $\boldsymbol{\gamma} = (\boldsymbol{\beta}', \boldsymbol{\theta}', \sigma^2)'$. Setting $\partial\ell/\partial\boldsymbol{\gamma} = 0$ gives the maximum likelihood equation. Specifically, a direct calculation of $\partial\ell/\partial\boldsymbol{\beta}$ yields the ML equation for $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\boldsymbol{Y} \,. \tag{38.4}$$

Denote by θ_k the kth element of the variance components $(\boldsymbol{\theta}, \sigma^2)$, where we label $\theta_{q+1} = \sigma^2$. Equating $\partial\ell/\partial\theta_k = 0$ gives

$$-\frac{1}{2}\left[\mathrm{tr}\left(\mathbf{V}^{-1}\frac{\partial\mathbf{V}}{\partial\theta_k}\right) - (\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta})' \times \mathbf{V}^{-1}\frac{\partial\mathbf{V}}{\partial\theta_k}\mathbf{V}^{-1}(\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta})\right] = 0 \,, \tag{38.5}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. In practice, iterations are required between (38.4) and (38.5) to obtain the MLEs. Furthermore the asymptotic sampling variance is routinely obtained from the inverse of the information matrix, which is minus the expected value of the matrix of second derivatives of the log-likelihood (38.3).

It is, however, worth pointing out that the MLEs obtained from (38.4, 5) are biased, especially for the variance components when the sample size is small. This is because the estimating equation (38.5) for the variance components fails to account for the loss of degrees of freedom when the true $\boldsymbol{\beta}$ is replaced by its estimate, $\hat{\boldsymbol{\beta}}$. To address this issue, an alternative maximum likelihood procedure, called the restricted maximum likelihood procedure, has been proposed for estimating the variance components [38.10]. The key idea is to replace the original response $\boldsymbol{Y}$ by a linear transform, so that the resulting 'response' contains no information about $\boldsymbol{\beta}$. The variance components can then be estimated based on this transformed response variable.

More specifically, choose a vector $\boldsymbol{a}$ such that $\boldsymbol{a}'\mathbf{X} = 0$. For more efficiency we use the maximum number, $\mathcal{N} - p$, of linearly independent vectors $\boldsymbol{a}$ and write $\mathbf{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{\mathcal{N}-p})$, which has a full row rank of $\mathcal{N} - p$. The restricted MLE will essentially apply the MLE procedure on $\mathbf{A}'\boldsymbol{Y}$, in lieu of the original $\boldsymbol{Y}$.

To proceed, we note that $\mathbf{A}'\boldsymbol{Y} \sim MVN(0, \mathbf{A}'\mathbf{V}\mathbf{A})$. The ML equations for the variance components can now be derived in a similar way to those for the original

$\boldsymbol{Y} \sim MVN(\mathbf{X}\boldsymbol{\beta}, \mathbf{V})$, namely by replacing $\boldsymbol{Y}$, $\mathbf{X}$ and $\mathbf{V}$ with $\mathbf{A}'\boldsymbol{Y}$, 0 and $\mathbf{AVA}'$ respectively in (38.5).

Caution must be exercised if the MLEs or the RMLEs of the variance components fall outside of the parameter space, as in the case of a negative estimate for a variance, in which case those solutions must be adjusted to yield estimates in the parameter space; see a more detailed discussion in *McCulloch* and *Searle* [38.11].

38.2.2 Prediction of Random Effects

A fixed effect differs from a random effect in that the former is considered to be constant and is often the main parameter we wish to estimate. In contrast, a random effect is considered to be an effect deriving from a population of effects. Consider again the aforementioned study of beam strength. Aside from the differences between the beams made from iron and alloy, there should be at least two sources of variability: (1) among beams produced by the same manufacturer; (2) between manufacturers. A simple random effects model can be specified as

$$E(Y_{ij}|b_i) = X_{ij}\beta + b_i \,,$$

where Y_{ij} is the strength of the j-th beam produced by the ith manufacturer and X_{ij} indicates whether iron or alloy was used to produce such a beam. Note that b_i is the effect on the strength of the beams produced by the i-th manufacturer, and this manufacturer was just the one among the selected manufacturers that happened to be labeled i in the study. The manufacturers were randomly selected as representative of the population of all manufacturers in the nation, and inferences about random effects were to be made about that population. Hence, estimating the variance components is of substantial interest for this purpose. On the other hand, one may wish to gain information about the performance of particular manufacturers. For instance, one may want to rank various manufacturers in order to select the best (or worst) ones. In these cases we are interested in predicting b_i.

In general the 'best' prediction of $\boldsymbol{b}$ in (38.2) based on observed response $\boldsymbol{Y}$ is required to minimize the mean squared error

$$\int (\hat{\boldsymbol{b}} - \boldsymbol{b})' \mathbf{G} (\hat{\boldsymbol{b}} - \boldsymbol{b}) f(\boldsymbol{Y}, \boldsymbol{b}) \mathrm{d}\boldsymbol{Y} \mathrm{d}\boldsymbol{b} \,, \tag{38.6}$$

where the predictor $\hat{\boldsymbol{b}}$ depends only on $\boldsymbol{Y}$, $f(\boldsymbol{Y}, \boldsymbol{b})$ is the joint density function of $\boldsymbol{Y}$ and $\boldsymbol{b}$, and $\mathbf{G}$ is a given nonrandom positive definite matrix. It can be shown for any given $\mathbf{G}$ that the minimizer is $E(\boldsymbol{b}|\boldsymbol{Y})$; in other words the conditional expectation of $\boldsymbol{b}$ given the observed response $\boldsymbol{Y}$.

If the variance components are known, an analytical solution exists based on the linear mixed model (38.2). That is, assuming $\boldsymbol{Y}$ and $\boldsymbol{b}$ follow a joint multinormal distribution, it follows that

$$E(\boldsymbol{b}|\boldsymbol{Y}) = \mathbf{DZ}'\mathbf{V}^{-1}(\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta}) \,.$$

Replacing $\boldsymbol{\beta}$ by its MLE

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\boldsymbol{Y}$$

would yield the Best linear unbiased predictor (BLUP) of random effects [38.12]. Because $\mathbf{D}$ and $\mathbf{V}$ are usually unknown, they are often replaced by their MLEs or RMLEs when calculating the BLUP, namely

$$\hat{\boldsymbol{b}} = \hat{\mathbf{D}}\hat{\mathbf{Z}}'\hat{\mathbf{V}}^{-1}(\boldsymbol{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \,.$$

Extensive derivation for the variance of the BLUP when the variance components are known has been given by *Henderson* et al. [38.12]. The variance of the BLUP with unknown variance components is not yet fully available.

38.3 Generalized Linear Mixed Models

Nonnormal data frequently arise from engineering studies. Consider again the beam study, where we now change the response to be a binary variable, indicating whether a beam has satisfied the criteria of quality control. For such nonnormal data, statistical models can be traced back to as early as 1934, when *Bliss* [38.13] proposed the first probit regression model for binary data. However, it took another four decades before *Nelder* and *Wedderburn* [38.1] and *McCullagh* and *Nelder* [38.2] proposed generalized linear models (GLMs) that could unify models and modeling techniques for analyzing more general data (such as counted data and polytomous data). Several authors [38.3–5] have considered a natural generalization of the GLMs to accommodate correlated nonnormal data. Their approach was to add random terms to the linear predictor parts, and the resulting models are termed generalized linear mixed models (GLMMs).

As an example, let $\boldsymbol{Y_{ij}}$ denote the status (such as a pass or a fail from the quality assurance test) of the

jth beam from the i-th manufacturer. We might create a model such as

$$\begin{aligned} Y_{ij}|b_i &\overset{\text{iid}}{\sim} \text{Bernoulli}\left(\mu_{ij}^{\text{b}}\right) ; \\ & i = 1, \ldots, m,\ j = 1, \ldots, n_i , \\ \text{logit}\left(\mu_{ij}^{\text{b}}\right) &= \boldsymbol{X}_{ij}'\boldsymbol{\beta} + b_i , \\ b_i &\overset{\text{iid}}{\sim} N\left(0, \sigma_u^2\right) , \end{aligned}$$

where $\text{logit}(\mu) = \log[\mu/(1-\mu)]$ is the link function that bridges the conditional probability and the linear predictors. The normal assumption for the random effects b_i is reasonable because the logit link carries the range of parameter space of μ_{ij} from $[0, 1]$ into the whole real line. Finally, we use independent b_i's to model the independent cluster effects and the within-cluster correlations among observations.

It is straightforward to generalize the above formulation to accommodate more general data. Specifically, let $\boldsymbol{X}_{ij}$ be a $p \times 1$ covariate vector associated with response Y_{ij}. Conditional on an unobserved cluster-specific random variable $\boldsymbol{b}_i$ (an $r \times 1$ vector), Y_{ij} are independent and follow a distribution of exponentials, that is

$$Y_{ij}|\boldsymbol{b}_i \overset{\text{iid}}{\sim} f(Y_{ij}|\boldsymbol{b}_i) , \tag{38.7}$$

$$f(Y_{ij}|\boldsymbol{b}_i) = \exp\left\{[Y_{ij}\alpha_{ij} - h(\alpha_{ij})]/\tau^2 - c(Y_{ij}, \tau)\right\} . \tag{38.8}$$

The conditional mean of $Y_{ij}|\boldsymbol{b}_i$, μ_{ij}^{b}, is related to α_{ij} through the identity $\mu_{ij}^{\text{b}} = \partial h(\alpha_{ij})/\partial\alpha_{ij}$, the transformation of which is to be modeled as a linear model in both the fixed and random effects:

$$g\left(\mu_{ij}^{\text{b}}\right) = \boldsymbol{X}_{ij}'\boldsymbol{\beta} + \boldsymbol{Z}_{ij}'\boldsymbol{b}_i , \tag{38.9}$$

where $g(\cdot)$ is termed a *link function*, often chosen to be invertible and continuous, and $\boldsymbol{Z}_{ij}$ is an $r \times 1$ design vector associated with the random effect. The random effects $\boldsymbol{b}_i$ are mutually independent with a common underlying distribution $F(\cdot; \boldsymbol{\theta})$ [or density $f(\cdot; \boldsymbol{\theta})$], where the variance components $\boldsymbol{\theta}$ is an unknown scalar or vector.

Model (38.9) is comprehensive and encompasses a variety of models. For continuous outcome data, by setting

$$h(\alpha) = \frac{1}{2}\alpha^2, \quad c(y, \tau^2) = \frac{1}{2}y^2/\tau^2 - \frac{1}{2}\log(2\pi\tau^2)$$

and $g(\cdot)$ to be an identity function, model (38.9) reduces to a linear mixed model. For binary outcome data, let

$$h(\alpha) = \log[1 + \exp(\alpha)] .$$

Choosing $g(\mu) = \text{logit}(\mu)$ yields a logit random effects model, while choosing $g(\mu) = \Phi^{-1}(\mu)$, where $\Phi(\cdot)$ is the CDF for a standard normal, gives a probit random effects model.

From (38.7) and (38.8) it is easy to construct the likelihood that the inference will be based on. That is,

$$\ell = \sum_{i=1}^{m} \log \int \prod_{j=1}^{n_i} f(Y_{ij}|\boldsymbol{b}_i; \boldsymbol{\beta}) f(\boldsymbol{b}_i; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b}_i ,$$

where the integration is over the r-dimensional random effect $\boldsymbol{b}_i$ and the summation results from independence across clusters.

We can also reformulate model (38.9) in a compact form that encompasses all of the data from all of the clusters. Using $\boldsymbol{Y}$, $\mathbf{X}$, $\mathbf{Z}$, $\boldsymbol{b}$ as defined in the previous section, we write

$$g[E(\boldsymbol{Y}|\boldsymbol{b})] = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{b} . \tag{38.10}$$

Hence, the log-likelihood function can be rewritten as

$$\begin{aligned} \ell(\boldsymbol{Y}; \boldsymbol{\beta}, \boldsymbol{\theta}) &= \log L(\boldsymbol{Y}; \boldsymbol{\beta}, \boldsymbol{\theta}) \\ &= \log \int f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b} , \end{aligned} \tag{38.11}$$

where $f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta})$ is the conditional likelihood for $\boldsymbol{Y}$ and $f(\boldsymbol{b}; \boldsymbol{\theta})$ is the density function for $\boldsymbol{b}$, often assumed to have a mean of zero.

Model (38.10) is not a simple reformat – it accommodates more complex data structures than clustered data. For example, with a properly defined $\mathbf{Z}$ and random effects $\boldsymbol{b}$ it encompasses crossed factor data [38.8] and nonnormal spatial data [38.14]. Hence, for more generality, the inferential procedures that we encounter in Sect. 38.4 will be based on (38.10, 11).

The GLMM is advantageous when the objective is to make inferences about individuals rather than the population average. Within its framework, random effects can be estimated and each individual's profile or growth curve can be obtained. The best predictor of random effects minimizing (38.6) is $E(\boldsymbol{Y}|\boldsymbol{b})$, which is not necessarily linear in $\boldsymbol{Y}$. However, if we confine our interest to the predictors that are linear in $\boldsymbol{Y}$, or of the form

$$\hat{\boldsymbol{b}} = \boldsymbol{c} + \mathbf{Q}\boldsymbol{Y}$$

for some conformable vector $\boldsymbol{c}$ and matrix $\mathbf{Q}$, minimizing the mean squared error (38.6) with respect to $\boldsymbol{c}$ and $\mathbf{Q}$ leads to the best *linear* predictor

$$\hat{\boldsymbol{b}} = E(\boldsymbol{b}) + \text{cov}(\boldsymbol{b}, \boldsymbol{Y})\text{var}(\boldsymbol{Y})[\boldsymbol{Y} - E(\boldsymbol{Y})] , \tag{38.12}$$

which holds true without any normality assumptions [38.11].

For example, consider a beta-binomial model for clustered binary outcomes such that

$$Y_{ij}|b_i \sim \text{Bernoulli}(b_i)$$

and the random effect $b_i \sim \text{Beta}(\alpha, \eta)$, where $\alpha, \eta > 0$. Using (38.12) we obtain the best linear predictor for b_i,

$$\hat{b}_i = \frac{\alpha + \bar{Y}_i}{\alpha + \beta + 1} ,$$

where $\bar{Y}_i = \sum_{j=1}^{n_i} Y_{ij}/n_i$.

38.4 Computing MLEs for GLMMs

A common theme when fitting a GLMM has been the difficulty involved with computating likelihood-based inference. Indeed, computing the likelihood itself is often challenging for GLMMs, mostly because of intractable integrals. This section presents various commonly used likelihood-based approaches to estimating the coefficients and variance components in GLMMs.

38.4.1 The EM Approach

The EM algorithm [38.15] is a widely used approach to calculating MLEs with missing observations. The basic idea behind its application to the random effects models is to treat the random terms as 'missing' data, and to impute the missing information based on the observed data. Imputations are often made via conditional expectations.

When drawing inference, our goal is to maximize the marginal likelihood of the observed data in order to obtain the MLEs for unknown $\boldsymbol{\beta}$ and variance components $\boldsymbol{\theta}$. If random effects ($\boldsymbol{b}$) were observed, we would be able to write the 'complete' data as ($\boldsymbol{Y}, \boldsymbol{b}$) with a joint log-likelihood

$$\ell(\boldsymbol{Y}, \boldsymbol{b}; \boldsymbol{\beta}, \boldsymbol{\theta}) = \log f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) + \log f(\boldsymbol{b}; \boldsymbol{\theta}) . \tag{38.13}$$

However, since $\boldsymbol{b}$ is unobservable, directly computing (38.13) is not feasible. Instead, the EM algorithm adopts a two-step iterative process. The expectation step ("E" step) computes the expectation of (38.13) conditional on the observed data. That is,

$$\tilde{\ell} = E\{\ell(\boldsymbol{Y}, \boldsymbol{b}; \boldsymbol{\beta}, \boldsymbol{\theta})|\boldsymbol{Y}, \boldsymbol{\beta}_0, \boldsymbol{\theta}_0\} ,$$

where $\boldsymbol{\beta}_0, \boldsymbol{\theta}_0$ are the current values, followed by the maximization step ("M" step), which maximizes $\tilde{\ell}$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. The E and M steps are iterated until convergence is achieved. Generally, the E step is computationally intensive, because it still needs to calculate a high-dimensional integral.

Indeed, since the conditional distribution of $\boldsymbol{b}|\boldsymbol{Y}$ involves the marginal distribution $f(\boldsymbol{Y})$, which is an intractable integral, a direct Monte Carlo simulation cannot fulfill the expectation step. In view of this difficulty, *McCulloch* [38.7] utilized the Metropolis–Hastings algorithm to make random draws from $\boldsymbol{b}|\boldsymbol{Y}$ without calculating the marginal density $f(\boldsymbol{Y})$.

The Metropolis–Hastings algorithm, dated back to the papers by *Metropolis* et al. [38.16] and *Hastings* [38.17], can be summarized as follows. Choose an auxiliary function $q(\boldsymbol{u}, \boldsymbol{v})$ such that $q(., \boldsymbol{v})$ is a pdf for all $\boldsymbol{v}$. This function is often called a *jumping distribution* from point $\boldsymbol{v}$ to $\boldsymbol{u}$. Draw $\boldsymbol{b}^*$ from $q(., \boldsymbol{b})$, where $\boldsymbol{b}$ is the current value of the Markov chain. Compute the ratio of importance

$$\omega = \frac{f(\boldsymbol{b}|\boldsymbol{Y})q(\boldsymbol{b}^*, \boldsymbol{b})}{f(\boldsymbol{b}^*|\boldsymbol{Y})q(\boldsymbol{b}, \boldsymbol{b}^*)} .$$

Set the current value of the Markov chain as $\boldsymbol{b}^*$ with probability $\min(1, \omega)$, and $\boldsymbol{b}$ has a probability $\max(0, 1 - \omega)$. It can be shown that, under mild conditions, the distribution of $\boldsymbol{b}$ drawn from such a procedure converges weakly to $f(\boldsymbol{b}|\boldsymbol{Y})$ (see, for example, [38.18]). Since the unknown density $f(\boldsymbol{Y})$ cancels out in the calculation of ω, the Metropolis–Hastings algorithm has successfully avoided computing $f(\boldsymbol{Y})$.

The ideal Metropolis–Hastings algorithm jumping rule is to sample the point directly from the target distribution. That is, in our case, $q(\boldsymbol{b}^*, \boldsymbol{b}) = f(\boldsymbol{b}^*|\boldsymbol{Y})$ for all $\boldsymbol{b}$. Then the ratio of importance, ω, is always 1, and the iterations of $\boldsymbol{b}^*$ are a sequence of independent draws from $f(\boldsymbol{b}^*|\boldsymbol{Y})$. In general, however, iterative simulation is applied to situations where direct sampling is not possible. Efficient jumping rules have been addressed by *Gelman* et al. [38.19].

We can now turn to the Monte Carlo EM algorithm, which takes the following form.

1. Choose initial values $\boldsymbol{\beta}^0$ and $\boldsymbol{\theta}^0$.

2. Denote the updated value at iteration s by $(\boldsymbol{\beta}^s, \boldsymbol{\theta}^s)$. Generate n values of $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^n$ from $f(\boldsymbol{b}|\boldsymbol{Y}; \boldsymbol{\beta}^s, \boldsymbol{\theta}^s)$.
3. At iteration $s+1$, choose $\boldsymbol{\beta}^{s+1}$ to maximize $\frac{1}{n}\sum_{k=1}^{n} \log f(\boldsymbol{Y}|\boldsymbol{b}^k; \boldsymbol{\beta})$.
4. Find $\boldsymbol{\theta}^{s+1}$ to maximize $\frac{1}{n}\sum_{k=1}^{n} \log f(\boldsymbol{b}^k; \boldsymbol{\theta})$.
5. Repeat steps 2–4 until convergence.

While computationally intensive, this algorithm is relatively stable since the log marginal likelihood increases at each iteration step and it is convergent at a linear rate [38.15].

38.4.2 Simulated Maximum Likelihood Estimation

Implementation of the EM is often computationally intensive. A naive approach would be to numerically approximate the likelihood (38.11) and maximize it directly. For example, when the random effects ($\boldsymbol{b}$) follow a normal distribution, we may use Gaussian quadrature to evaluate (38.11) and its derivatives. However, this approach quickly fails when the dimensions of $\boldsymbol{b}$ are large. We now consider a simulation technique, namely, simulated maximum likelihood estimation, to approximate the likelihood directly and, further, to obtain the MLEs. The key idea behind this approach is to approximate (38.11) and its first- and second-order derivatives by Monte Carlo simulations while performing Newton-Raphson iterations.

We begin with the likelihood approximation. Following *Geyer* and *Thompson* [38.20] and *Gelfand* and *Carlin* [38.21], one notices that for any density function $h(\boldsymbol{b})$ with the same support as $f(\boldsymbol{b}; \boldsymbol{\theta})$,

$$L(\boldsymbol{Y}; \boldsymbol{\beta}, \boldsymbol{\theta}) = \int \frac{f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta})}{h(\boldsymbol{b})} h(\boldsymbol{b}) \mathrm{d}\boldsymbol{b} \,. \quad (38.14)$$

Hence, Monte Carlo simulations can be applied to evaluate $L(\boldsymbol{Y}; \boldsymbol{\beta}, \boldsymbol{\theta})$. Explicitly, if $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^n$ are generated independently from $h(\boldsymbol{b})$ (termed an *importance sampling distribution*), (38.14) can be approximated by

$$1/n \sum_{i=1}^{n} \frac{f(\boldsymbol{Y}|\boldsymbol{b}^i; \boldsymbol{\beta}) f(\boldsymbol{b}^i; \boldsymbol{\theta})}{h(\boldsymbol{b}^i)} \quad (38.15)$$

with an accuracy of order $O_p(n^{-1/2})$. The optimal (in the sense that the Monte Carlo approximation has zero variance) importance sampling distribution is $f(\boldsymbol{b}|\boldsymbol{Y})$, evaluated at the MLEs [38.22]. However, since the MLEs are unknown and the conditional distribution cannot be evaluated, such an optimal distribution is never meaningful practically. Nevertheless, we can find a distribution (such as a normal distribution) to approximate $f(\boldsymbol{b}|\boldsymbol{Y})$.

More specifically, notice that

$$\begin{aligned} f(\boldsymbol{b}|\boldsymbol{Y}) &= c \times f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \\ &= c \times \exp[-K(\boldsymbol{Y}, \boldsymbol{b})] \,, \end{aligned}$$

where c (which does not depend on $\boldsymbol{b}$) is used to ensure a proper density function. We use

$$\begin{aligned} h(\boldsymbol{b}; \boldsymbol{\beta}, \boldsymbol{\theta}) =& ||2\pi \hat{\boldsymbol{\Sigma}}||^{-1/2} \\ & \times \exp\left[-\frac{1}{2}(\boldsymbol{b} - \hat{\boldsymbol{b}})' \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{b} - \hat{\boldsymbol{b}})\right] \,, \end{aligned}$$

where $||\cdot||$ denotes the determinant of a square matrix, $\hat{\boldsymbol{b}} = \mathrm{argmin}_{\boldsymbol{b}}[K(\boldsymbol{Y}, \boldsymbol{b})]$ and $\hat{\boldsymbol{\Sigma}} = [\frac{\partial}{\partial \boldsymbol{b} \partial \boldsymbol{b}'} K(\boldsymbol{Y}, \hat{\boldsymbol{b}})]^{-1}$, to approximate the conditional density $f(\boldsymbol{b}|\boldsymbol{Y})$ evaluated at $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. Similarly, the derivatives of $L(\boldsymbol{Y}; \boldsymbol{\beta}, \boldsymbol{\theta})$ can also be approximated by Monte Carlo simulations.

Then the algorithm proceeds as follows:

1. Choose the initial values $\boldsymbol{\gamma}^0 = (\boldsymbol{\beta}^0, \boldsymbol{\theta}^0)$ for $\boldsymbol{\gamma} = (\boldsymbol{\beta}, \boldsymbol{\theta})$.
2. Denote the current value at the sth step by $\boldsymbol{\gamma}^s$. Generate $\boldsymbol{b}^1, \ldots, \boldsymbol{b}^n$ based on $h(\boldsymbol{b}|\boldsymbol{\gamma}^s)$.
3. Calculate the approximate derivatives of the marginal likelihood function $L(\boldsymbol{Y}; \boldsymbol{\beta}, \boldsymbol{\theta})$ evaluated at $\boldsymbol{\gamma}^s$:

$$\mathcal{B}_\beta^s = \frac{1}{n}\sum_{k=1}^{n} \frac{f(\boldsymbol{b}^k; \boldsymbol{\theta}^s)}{h(\boldsymbol{b}^k; \boldsymbol{\gamma}^s)} \frac{\partial}{\partial \boldsymbol{\beta}} f(\boldsymbol{Y}|\boldsymbol{b}^k; \boldsymbol{\beta})|_{\boldsymbol{\beta}^s} \,,$$

$$\mathcal{B}_\theta^s = \frac{1}{n}\sum_{k=1}^{n} \frac{f(\boldsymbol{Y}|\boldsymbol{b}^k; \boldsymbol{\beta}^s)}{h(\boldsymbol{b}^k; \boldsymbol{\gamma}^s)} \frac{\partial}{\partial \boldsymbol{\theta}} f(\boldsymbol{b}^k; \boldsymbol{\theta})|_{\boldsymbol{\beta}^s} \,,$$

$$\mathcal{A}_{\beta\beta}^s = \frac{1}{n}\sum_{k=1}^{n} \frac{f(\boldsymbol{b}^k; \boldsymbol{\theta}^s)}{h(\boldsymbol{b}^k; \boldsymbol{\gamma}^s)} \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} f(\boldsymbol{Y}|\boldsymbol{b}^k; \boldsymbol{\beta})|_{\boldsymbol{\beta}^s} \,,$$

$$\mathcal{A}_{\theta\theta}^s = \frac{1}{n}\sum_{k=1}^{n} \frac{f(\boldsymbol{Y}|\boldsymbol{b}^k, \boldsymbol{\beta}^s)}{h(\boldsymbol{b}^k; \boldsymbol{\gamma}^s)} \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} f(\boldsymbol{b}^k; \boldsymbol{\theta})|_{\boldsymbol{\theta}^s} \,,$$

$$\begin{aligned} \mathcal{A}_{\beta\theta}^s =& \frac{1}{n}\sum_{k=1}^{n} \frac{1}{h(\boldsymbol{b}^k; \boldsymbol{\gamma}^s)} \frac{\partial}{\partial \boldsymbol{\beta}} f(\boldsymbol{Y}|\boldsymbol{b}^k; \boldsymbol{\beta})|_{\boldsymbol{\beta}^s} \\ & \times \left[\frac{\partial}{\partial \boldsymbol{\theta}} f(\boldsymbol{b}^k; \boldsymbol{\theta})|_{\boldsymbol{\theta}^s}\right]' \,. \end{aligned}$$

4. Compute the updated value at the $(s+1)$th step

$$\boldsymbol{\gamma}^{s+1} = \boldsymbol{\gamma}^s - (\mathcal{A}^s)^{-1} \mathcal{B}^s \,,$$

$$\text{where } \mathcal{A}^s = \begin{pmatrix} \mathcal{A}_{\beta\beta}^s & \mathcal{A}_{\beta\theta}^s \\ \left(\mathcal{A}_{\beta\theta}^s\right)' & \mathcal{A}_{\theta\theta}^s \end{pmatrix}$$

$$\text{and } \mathcal{B}^s = \left(\mathcal{B}_\beta^{s\prime}, \mathcal{B}_\theta^{s\prime}\right)' \,.$$

5. Repeat steps 2–4 until convergent criteria are met. Upon convergence, set $\hat{\boldsymbol{\gamma}} = \boldsymbol{\gamma}^s$ and the Hessian matrix $\mathcal{A} = \mathcal{A}^s$.

The covariance of the resulting $\hat{\boldsymbol{\gamma}}$ is approximated (ignoring the Monte Carlo error) by the inverse of the observed information matrix, given by

$$-\frac{\partial^2}{\partial\boldsymbol{\gamma}\partial\boldsymbol{\gamma}'}\log L(\boldsymbol{Y};\boldsymbol{\beta},\boldsymbol{\theta})|_{\hat{\boldsymbol{\gamma}}} \doteq -\hat{L}^{-1}\mathcal{A}\,,$$

where $\hat{L}$ and $\mathcal{A}$ are the approximations of $L(\boldsymbol{Y};\boldsymbol{\beta},\boldsymbol{\theta})$ and the Hessian matrix evaluated at $\hat{\boldsymbol{\gamma}} = (\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\theta}})$, respectively.

38.4.3 Monte Carlo Newton–Raphson (MCNR)/ Stochastic Approximation (SA)

Monte Carlo Newton-Raphson and stochastic approximation are two similar approaches to finding the MLEs for the GLMMs. They both approximate the score function using simulated random effects and improve the precision of the approximation at each iteration step.

We first describe a typical (MCNR) algorithm. Consider the decomposition of the joint density of the response vector and random effects vector

$$f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma}) = f(\boldsymbol{Y};\boldsymbol{\gamma})f(\boldsymbol{b}|\boldsymbol{Y};\boldsymbol{\gamma})\,.$$

Hence

$$\frac{\partial\log f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}} = S(\boldsymbol{\gamma}) + \frac{\partial\log f(\boldsymbol{b}|\boldsymbol{Y};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}\,, \quad (38.16)$$

where $S(\boldsymbol{\gamma}) = \partial\log f(\boldsymbol{Y};\boldsymbol{\gamma})/\partial\boldsymbol{\gamma}$, the score function of main interest. In view of

$$E\left(\frac{\partial\log f(\boldsymbol{b}|\boldsymbol{Y};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}|\boldsymbol{Y}\right) = 0\,,$$

(38.16) can be written in the format of a regression equation

$$\frac{\partial\log f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}} = S(\boldsymbol{\gamma}) + \text{error}\,,$$

where the "error" term substitutes $\partial\log f(\boldsymbol{b}|\boldsymbol{Y};\boldsymbol{\gamma})/\partial\boldsymbol{\gamma}$, a mean zero term. Thus, inserting values of $\boldsymbol{b}\sim f(\boldsymbol{b}|\boldsymbol{Y})$ into $\partial\log f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma})/\partial\boldsymbol{\gamma}$ yields "data" to perform such a regression.

The MCNR algorithm is typically implemented as follows. Denote by $\boldsymbol{\gamma}^{(s)}$ the value of the estimate of $\boldsymbol{\gamma}$ at iteration step s. Generate via the Metropolis-Hastings algorithm a sequence of realized values $\boldsymbol{b}^{(s,1)},\ldots,\boldsymbol{b}^{(s,n)}\sim f(\boldsymbol{b}|\boldsymbol{Y};\boldsymbol{\gamma}^{(s)})$. At the $(s+1)$th step, compute

$$\boldsymbol{\gamma}^{(s+1)} = \boldsymbol{\gamma}^{(s)} - a_s\hat{E}\left(\frac{\partial\log f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^{(s)}}\right)\,. \quad (38.17)$$

Here a_s is a constant, incorporating information about the expectation of the derivative of $\partial\log f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma})/\partial\boldsymbol{\gamma}$ at the root, an unknown quantity. In practice, a_s is often set to be the inverse of a Monte Carlo estimate of the expectation based on the realized values of $\boldsymbol{b}^{(s,1)},\ldots,\boldsymbol{b}^{(s,n)}$.

The SA differs from the MCNR in that the SA uses a single simulated value of random effects in (38.17), that is

$$\boldsymbol{\gamma}^{(s+1)} = \boldsymbol{\gamma}^{(s)} - a_s\frac{\partial\log f(\boldsymbol{Y},\boldsymbol{b}^{(s)};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^{(s)}}\,,$$

and a_s is chosen to gradually decrease to zero. *Ruppert* and *Gu* and *Kong* have recommended that

$$a_s = \frac{e}{(s+\kappa)^\alpha}\left[\hat{E}\left(\frac{\partial^2\log f(\boldsymbol{Y},\boldsymbol{b};\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}\partial\boldsymbol{\gamma}'}\right)\right]^{-1}\,,$$

where $e=3$, $\kappa=50$ and $\alpha=0.75$ as chosen by *McCulloch* and *Searle* [38.11]. The multiplier a_s decreases the step size as the iterations increase in the SA and eventually serves to eliminate the stochastic error involved in the Metropolis-Hastings steps. *McCulloch* and *Searle* [38.11] stated that the SA is advantageous in that it can use all of the simulated data to calculate estimates and only uses the simulated values one at a time; however, the detailed implementations of both methods are yet to be settled on in the literature.

38.4.4 S–U Algorithm

The S–U algorithm is a technique for finding the solution of an estimating equation that can be expressed as the expected value of a full data estimating equation, where the expectation is taken with respect to the missing data, given the observed data. This algorithm alternates between two steps: a simulation step wherein the missing values are simulated based on the conditional distributions given the observed data, and an updating step wherein parameters are updated without performing a numerical maximization. An attractive feature of this approach is that it is sequential – the number of Monte Carlo replicates does not have to be specified in advance, and the values of previous Monte Carlo replicates do not have to be stored or regenerated for later use. In the following, we will apply this approach in order to solve the maximum likelihood equations.

Differentiating the log-likelihood (38.26) with respect to the unknown parameters, $\boldsymbol{\gamma} = (\boldsymbol{\beta}, \boldsymbol{\theta})$, gives

$$\boldsymbol{S}_b(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{\partial \ell}{\partial \boldsymbol{\beta}} = \frac{1}{f(\boldsymbol{Y}; \boldsymbol{\gamma})} \times \int \boldsymbol{S}_b(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b} \,, \tag{38.18}$$

$$\boldsymbol{S}_t(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{\partial \ell}{\partial \boldsymbol{\theta}} = \frac{1}{f(\boldsymbol{Y}; \boldsymbol{\gamma})} \times \int \boldsymbol{S}_t(\boldsymbol{b}; \boldsymbol{\theta}) f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b} \,, \tag{38.19}$$

where $f(\boldsymbol{Y}; \boldsymbol{\gamma})$ is the marginal likelihood of the observed data set, and $\boldsymbol{S}_b(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta})$, $\boldsymbol{S}_t(\boldsymbol{b}; \boldsymbol{\theta})$ are conditional scores when treating $\boldsymbol{b}$ as observed constants, that is $\boldsymbol{S}_b(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) = \partial \log f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta})/\partial \boldsymbol{\beta}$, and $\boldsymbol{S}_t(\boldsymbol{b}; \boldsymbol{\theta}) = \partial \log f(\boldsymbol{b}; \boldsymbol{\theta})/\partial \boldsymbol{\theta}$.

Some algebra gives the second derivatives of the log-likelihood, which are needed in the algorithm. More specifically,

$$\boldsymbol{S}_{bb}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} = -\boldsymbol{S}_b^{\otimes 2}(\boldsymbol{\beta}, \boldsymbol{\theta}) + \frac{1}{f(\boldsymbol{Y}; \boldsymbol{\gamma})} \times \int \{\boldsymbol{S}_{bb}(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) + \boldsymbol{S}_b^{\otimes 2}(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta})\} f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b} \,, \tag{38.20}$$

$$\boldsymbol{S}_{bt}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\theta}'} = -\boldsymbol{S}_b(\boldsymbol{\beta}, \boldsymbol{\theta}) \boldsymbol{S}_t'(\boldsymbol{\beta}, \boldsymbol{\theta}) + \frac{1}{f(\boldsymbol{Y}; \boldsymbol{\gamma})} \times \int \boldsymbol{S}_b(\boldsymbol{\beta}, \boldsymbol{\theta}) \boldsymbol{S}_t'(\boldsymbol{b}; \boldsymbol{\theta}) f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b} \,, \tag{38.21}$$

$$\boldsymbol{S}_{tt}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{\partial^2 \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} = -\boldsymbol{S}_t^{\otimes 2}(\boldsymbol{\beta}, \boldsymbol{\theta}) + \frac{1}{f(\boldsymbol{Y}; \boldsymbol{\gamma})} \times \int \left\{\boldsymbol{S}_{tt}(\boldsymbol{b}; \boldsymbol{\theta}) + \boldsymbol{S}_t^{\otimes 2}(\boldsymbol{b}; \boldsymbol{\theta})\right\} f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) f(\boldsymbol{b}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{b} \,, \tag{38.22}$$

where $\boldsymbol{S}_{bb}(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta})$, $\boldsymbol{S}_{tt}(\boldsymbol{b}; \boldsymbol{\theta})$ are conditional information when treating $\boldsymbol{b}$ as observed constants, that is $\boldsymbol{S}_{bb}(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta}) = \partial^2 \log f(\boldsymbol{Y}|\boldsymbol{b}; \boldsymbol{\beta})/\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'$, and $\boldsymbol{S}_{tt}(\boldsymbol{b}; \boldsymbol{\theta}) = \partial^2 \log f(\boldsymbol{b}; \boldsymbol{\theta})/\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'$. Here for a column vector $\boldsymbol{a}$, $\boldsymbol{a}^{\otimes 2} = \boldsymbol{a}\boldsymbol{a}'$.

Hence, one can use the importance sampling scheme [38.23] to approximate these functions and their derivatives. We proceed as follows.

Having obtained the approximants $\hat{\boldsymbol{\gamma}}_1 = (\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\theta}}_1), \ldots, \hat{\boldsymbol{\gamma}}_j = (\hat{\boldsymbol{\beta}}_j, \hat{\boldsymbol{\theta}}_j)$ to $\hat{\boldsymbol{\gamma}} = (\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}})$, the true MLE, at the jth S-step of the algorithm, we simulate $\boldsymbol{b}^{(j,l)}, l = 1, \ldots, n$, independently from $f(\boldsymbol{b}; \hat{\boldsymbol{\theta}}_j)$. Denote $w^{(j,l)}$ by

$$w^{(j,l)} = f\left(\boldsymbol{Y}|\boldsymbol{b}^{(j,l)}; \hat{\boldsymbol{\beta}}_j\right)$$

and let

$$\bar{w}_j = \frac{1}{j \cdot n} \sum_{j'=1}^{j} \sum_{l=1}^{n} w^{(j',l)} \,.$$

As $j \to \infty$, the law of large numbers gives that $\bar{w}_j$ is asymptotically equal to $f(\boldsymbol{Y}; \hat{\boldsymbol{\gamma}})$ provided that $\hat{\boldsymbol{\gamma}}_j \xrightarrow{p} \hat{\boldsymbol{\gamma}}$.

We write

$$\bar{\boldsymbol{S}}_{b,j} = \frac{1}{jn\bar{w}_j} \sum_{j'=1}^{j} \sum_{l=1}^{n} w^{(j',l)} \boldsymbol{S}_b\left(\boldsymbol{Y}|\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right) \,,$$

$$\bar{\boldsymbol{S}}_{t,j} = \frac{1}{jn\bar{w}_j} \sum_{j'=1}^{j} \sum_{l=1}^{n} w^{(j',l)} \boldsymbol{S}_t\left(\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\theta}}_j\right) \,,$$

$$\bar{\boldsymbol{S}}_{bb,j} = -\bar{\boldsymbol{S}}_{b,j}^{\otimes 2} + \frac{1}{jn\bar{w}_j} \sum_{j'=1}^{j} \sum_{l=1}^{n} w^{(j',l)} \times \left[\boldsymbol{S}_{bb}\left(\boldsymbol{Y}|\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right) + \boldsymbol{S}_b^{\otimes 2}\left(\boldsymbol{Y}|\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right)\right] \,,$$

$$\bar{\boldsymbol{S}}_{tt,j} = -\bar{\boldsymbol{S}}_{t,j}^{\otimes 2} + \frac{1}{jn\bar{w}_j} \sum_{j'=1}^{j} \sum_{l=1}^{n} w^{(j',l)} \times \left[\boldsymbol{S}_{tt}\left(\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right) + \boldsymbol{S}_t^{\otimes 2}\left(\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right)\right] \,,$$

$$\bar{\boldsymbol{S}}_{bt,j} = -\bar{\boldsymbol{S}}_{b,j} \bar{\boldsymbol{S}}_{t,j}' + \frac{1}{jn\bar{w}_j} \sum_{j'=1}^{j} \sum_{l=1}^{n} w^{(j',l)} \times \left[\boldsymbol{S}_b\left(\boldsymbol{Y}|\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right) \boldsymbol{S}_t'\left(\boldsymbol{b}^{(j',l)}; \hat{\boldsymbol{\beta}}_j\right)\right] \,,$$

With j sufficiently large, $\bar{\boldsymbol{S}}_{b,j}, \bar{\boldsymbol{S}}_{t,j}, \bar{\boldsymbol{S}}_{bb,j}, \bar{\boldsymbol{S}}_{bt,j}, \bar{\boldsymbol{S}}_{tt,j}$ provide good estimates for (38.18, 22).

Denote $\boldsymbol{S}_j = (\boldsymbol{S}_{b,j}', \boldsymbol{S}_{t,j}')'$ and

$$\boldsymbol{H}_j = \begin{pmatrix} \bar{\boldsymbol{S}}_{bb,j} & \bar{\boldsymbol{S}}_{bt,j} \\ \bar{\boldsymbol{S}}_{bt,j}' & \bar{\boldsymbol{S}}_{tt,j} \end{pmatrix} \,.$$

Then, at the jth U-step, the updated value for $\hat{\boldsymbol{\gamma}}$ is

$$\boldsymbol{\gamma}^{(j+1)} = \boldsymbol{\gamma}^{(j)} - a_j \boldsymbol{H}_j^{-1} \boldsymbol{S}_j \,,$$

where the tuning parameter a_j can be chosen as discussed in the previous section. Note that each of the quantities required at this step, such as $\bar{\boldsymbol{S}}_j$, $\bar{\boldsymbol{S}}_{\boldsymbol{\beta},j}$, and so on, can be calculated recursively so that the past values of these intermediate variables never need to be stored.

Following *Satten* and *Datta* [38.24], as $j \to \infty$, $\hat{\boldsymbol{\gamma}}_j$ almost surely converges to $\hat{\boldsymbol{\gamma}}$. Denote the S–U estimate

by $\hat{\boldsymbol{\gamma}}_{\mathrm{SU}}$. The total sampling variance of $\hat{\boldsymbol{\gamma}}_{\mathrm{SU}}$ around $\boldsymbol{\gamma}_0$ is the sum of the variance of $\hat{\boldsymbol{\gamma}}_{\mathrm{SU}}$ around $\hat{\boldsymbol{\gamma}}$ due to the S–U algorithm and the sampling variance of $\hat{\boldsymbol{\gamma}}$ around $\boldsymbol{\gamma}_0$ [38.25]. In most cases, the S–U algorithm should be iterated until the former is negligible compared to the latter. In theory, the starting value for the S–U algorithm is arbitrary. However, a poor starting value might cause instability at the beginning of this algorithm. Hence, in the next section, we consider several approximate methods that generate a starting value sufficiently close to the true zero of the estimating equations.

38.4.5 Some Approximate Methods

In view of the cumbersome and often intractable integrations required for a full likelihood analysis, several techniques have been made available for approximate inference in the GLMMs and other nonlinear variance component models.

The penalized quasi-likelihood (PQL) method introduced by *Green* [38.26] for semiparametric models was initially exploited as an approximate Bayes procedure to estimate regression coefficients. Since then, several authors have used the PQL to draw approximate inferences based on random effects models: *Schall* [38.5] and *Breslow* and *Clayton* [38.8] developed iterative PQL algorithms, *Lee* and *Nelder* [38.27] applied the PQL directly to hierarchical models. We present the PQL from the likelihood perspective below.

Consider the GLMM (38.10). For notational simplicity, we write the integrand of the likelihood function

$$f(\boldsymbol{Y}|\boldsymbol{b};\boldsymbol{\beta})f(\boldsymbol{b};\boldsymbol{\theta}) = \exp[-K(\boldsymbol{Y},\boldsymbol{b})]\,. \tag{38.23}$$

More generally, if one only specifies the first two conditional moments of $\boldsymbol{Y}$ given $\boldsymbol{b}$ in lieu of a full likelihood specification, $f(\boldsymbol{Y}|\boldsymbol{b};\boldsymbol{\beta})$ in (38.23) can be replaced by the quasi-likehood function $\exp[ql(\boldsymbol{Y}|\boldsymbol{b};\boldsymbol{\beta})]$, where

$$ql(\boldsymbol{Y}|\boldsymbol{b};\boldsymbol{\beta}) = \sum_{i=1}^{m}\sum_{j=1}^{n_i}\int_{Y_{ij}}^{\mu_{ij}^{b}} \frac{Y_{ij}-t}{V(t)}\mathrm{d}\,t\,.$$

Here $\mu_{ij}^{\mathrm{b}} = E(Y_{ij}|\boldsymbol{b};\boldsymbol{\beta})$ and $V(\mu_{ij}^{\mathrm{b}}) = \mathrm{var}(Y_{ij}|\boldsymbol{b};\boldsymbol{\beta})$.

Next evaluate the marginal likelihood. We temporarily assume that $\boldsymbol{\theta}$ is known. For any fixed $\boldsymbol{\beta}$, expanding $K(\boldsymbol{Y},\boldsymbol{b})$ around its mode $\hat{\boldsymbol{b}}$ up to the second-order term, we have

$$\begin{aligned} L(\boldsymbol{Y};\boldsymbol{\beta},\boldsymbol{\theta}) &= \int \exp[-K(\boldsymbol{Y},\boldsymbol{b})]\mathrm{d}\,\boldsymbol{b} \\ &= \left|\left|2\pi\left[K''(\boldsymbol{Y},\tilde{\boldsymbol{b}})\right]^{-1}\right|\right|^{1/2}\exp\left[-K(\boldsymbol{Y},\hat{\boldsymbol{b}})\right], \end{aligned}$$

where $K''(\boldsymbol{Y},\boldsymbol{b})$ denotes the second derivative of $K(\boldsymbol{Y},\boldsymbol{b})$ with respect to $\boldsymbol{b}$, and $\tilde{\boldsymbol{b}}$ lies in the segment joining zero and $\hat{\boldsymbol{b}}$. If $K''(\boldsymbol{Y},\boldsymbol{b})$ does not vary too much as $\boldsymbol{b}$ changes (for instance, $K''(\boldsymbol{Y},\boldsymbol{b}) = \textit{constant}$ for normal data), maximizing the marginal likelihood (38.11) is equivalent to maximizing

$$\mathrm{e}^{-K(\boldsymbol{Y},\hat{\boldsymbol{b}})} = f(\boldsymbol{Y}|\hat{\boldsymbol{b}},\boldsymbol{\beta})f(\hat{\boldsymbol{b}};\boldsymbol{\theta})\,.$$

Or, equivalently, $\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})$ and $\hat{\boldsymbol{b}}(\boldsymbol{\theta})$ are obtained by jointly maximizing $f(\boldsymbol{Y}|\boldsymbol{b};\boldsymbol{\beta})f(\boldsymbol{b};\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\beta}$ and $\boldsymbol{b}$ with $\boldsymbol{\theta}$ being held constant. If $\boldsymbol{\theta}$ is unknown, it can be estimated by maximizing the approximate profile likelihood of $\boldsymbol{\theta}$,

$$\left|\left|2\pi\left\{K''[\boldsymbol{Y},\hat{\boldsymbol{b}}(\boldsymbol{\theta})]\right\}^{-1}\right|\right|^{1/2}\exp\left\{-K[\boldsymbol{Y},\hat{\boldsymbol{b}}(\boldsymbol{\theta})]\right\}.$$

A more detailed discussion can be found in *Breslow* and *Clayton* [38.8].

As no closed-form solution is available, the PQL is often performed through an iterative process. In particular, *Schall* [38.5] derived an iterative algorithm where the random effects follow normal distributions. Specifically, with the current estimated values of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and $\boldsymbol{b}$, a working 'response' $\tilde{\boldsymbol{Y}}$ is constructed by the first-order Taylor expansion of $g(\boldsymbol{Y})$ around $\boldsymbol{\mu}^{\mathrm{b}}$, or explicitly,

$$\begin{aligned} \tilde{\boldsymbol{Y}} &= g(\boldsymbol{\mu}^{\mathrm{b}}) + g'(\boldsymbol{\mu}^{\mathrm{b}})(\boldsymbol{Y}-\boldsymbol{\mu}^{\mathrm{b}}) \\ &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}b + g'(\boldsymbol{\mu}^{\mathrm{b}})(\boldsymbol{Y}-\boldsymbol{\mu}^{\mathrm{b}})\,, \end{aligned} \tag{38.24}$$

where $g(\cdot)$ is defined in (38.9).

Viewing the last term in (38.24) as a random error, (38.24) suggests fitting a linear mixed model on $\tilde{\boldsymbol{Y}}$ to obtain the updated values of $\boldsymbol{\beta}, \boldsymbol{b}$ and $\boldsymbol{\theta}$, which are used to recalculate the working 'response'. The iteration continues until convergence. Computationally, the PQL is easy to implement; it only requires repeatedly calling in existing software, for example, SAS 'PROC MIXED'. The PQL procedure yields exact MLEs for normally distributed data and for some cases when the conditional distribution of $\boldsymbol{Y}$ and the distribution of $\boldsymbol{b}$ are conjugate.

Other approaches, such as the Laplace method and the Solomon-Cox approximation, have also received much attention. The Laplace method (see for example *Liu* and *Pierce* [38.28]) differs from the PQL only in that the former obtains $\hat{\boldsymbol{b}}(\boldsymbol{\beta},\boldsymbol{\theta})$ by maximizing the integrand $\mathrm{e}^{-K(\boldsymbol{Y},\boldsymbol{b})}$ with $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ being held fixed, and subsequently estimates $(\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\theta}})$ by jointly maximizing

$$\left|\left|2\pi\left[K''(\boldsymbol{Y},\hat{\boldsymbol{b}})\right]^{-1}\right|\right|^{1/2}\exp\left[-K(\boldsymbol{Y},\hat{\boldsymbol{b}})\right].$$

On the other hand, with the assumption of $E(\boldsymbol{b}) = 0$, the Solomon-Cox technique approximates

the integral $\int f(\boldsymbol{Y}|\boldsymbol{b})f(\boldsymbol{b})\mathrm{d}\boldsymbol{b}$ by expanding the integrand $f(\boldsymbol{Y}|\boldsymbol{b})$ around $\boldsymbol{b}=0$; see *Solomon* and *Cox* [38.29].

In general, none of these approximate methods produce consistent estimates,h except in some special cases, for example with normal data. Moreover, these methods are essentially based on normal approximation, and they often do not perform well for sparse data, such as binary data, and when the cluster size is relatively small [38.30].

38.5 Special Topics: Testing Random Effects for Clustered Categorical Data

It is useful to test for correlation within clusters and the heterogeneity among clusters when (or prior to) fitting random effects models. Tests have been proposed that are based on score statistics for the null hypothesis that variance components are zero for clustered continuous, binary and Poisson outcomes within the random effects model framework [38.31, 32]. However, literature that deals with tests for clustered polytomous data is scarce.

A recent article by *Li* and *Lin* [38.33] investigated tests for within-cluster correlation for clustered polytomous and censored discrete time-to-event data by deriving score tests for the null hypothesis that variance components are zero in random effects models. Since the null hypothesis is on the boundary of the parameter space, unlike the Wald and likelihood ratio tests whose asymptotic distributions are mixtures of chi-squares, the score tests are advantageous because their asymptotic distributions are still chi-square. Another advantage of the score tests is that no distribution needs to be assumed for the random effects except for their first two moments. Hence they are robust to mis-specifying the distributions of the random effects. Further, the Wald tests and the LR tests involve fitting random effects models that involve numerical integration, in contrast with the score tests, which only involve fitting standard models under the null hypothesis using existing standard software, and do not require numerical integration.

A common problem in the analysis of clustered data is the presence of covariate measurement errors. For example, in flood forecasting studies, the radar measurements of precipitation are 'highly susceptible' to error due to improper electronic calibration [38.34]; in AIDS studies, CD4 counts are often measured with error [38.35]. Valid statistical inference needs to account for measurement errors in covariates. *Li* and *Lin* [38.33] have extended the score tests for variance components to the situation where covariates are measured with errors. They applied the SIMEX method [38.36] to correct for measurement errors and develop SIMEX score tests for variance components. These tests are an extension of the SIMEX score test of *Lin* and *Carroll* [38.37] to clustered polytomous data with covariate measurement error.

Random effects-generalized logistic models and cumulative probability models have been proposed to model clustered nominal and ordinal categorical data [38.38, 39]. This section focuses on the score tests for the null hypothesis that the variance components are zero in such models to test for the within-cluster correlation.

38.5.1 The Variance Component Score Test in Random Effects–Generalized Logistic Models

Suppose that, for the jth ($j=1,\ldots,n_i$) subject in the i-th ($i=1,\ldots,m$) cluster, a categorical response Y_{ij} belongs to one of N categories indexed by $1,\ldots,N$. Conditional on the cluster-level random effect b_i, the observations Y_{ij} are independent and the conditional probability $P_{ij,k}=P(Y_{ij}=k|b_i)$ depends on the $p\times 1$ covariate vector $\boldsymbol{X}_{ij}$ through a generalized logistic model

$$\log\left(\frac{P_{ij,k}}{P_{ij,N}}\right)=\alpha_k+\boldsymbol{X}'_{ij}\boldsymbol{\beta}_k+b_i=\boldsymbol{X}'_{ij,k}\boldsymbol{\beta}+b_i\,,\quad k=1,\ldots,N-1 \tag{38.25}$$

where $\boldsymbol{\beta}_k$ is a $p\times 1$ vector of fixed effects, $b_i\sim F(b_i;\theta)$ for some distribution function F that has zero mean and a variance θ, $\boldsymbol{X}'_{ij,k}=\boldsymbol{e}'_k\otimes(1,\boldsymbol{X}'_{ij})$; $\otimes$ denotes a Kronecker product, $\boldsymbol{e}_k$ is an $(N-1)\times 1$ vector with the kth component equal to 1 and the rest of the components set to zero, and $\boldsymbol{\beta}=(\alpha_1,\boldsymbol{\beta}'_1,\cdots,\alpha_{N-1},\boldsymbol{\beta}'_{N-1})'$.

The marginal log-likelihood function for $(\boldsymbol{\beta},\theta)$ is

$$\ell(\boldsymbol{\beta},\theta)=\sum_{i=1}^{m}\log\int\exp[\ell_i(\boldsymbol{\beta},b_i)]\mathrm{d}\,F(b_i;\theta)\,, \tag{38.26}$$

where $\ell_i(\boldsymbol{\beta},b_i)=\sum_{j=1}^{n_i}\sum_{k=1}^{N}y_{ij,k}\log P_{ij,k}$, $y_{ij,k}=I(Y_{ij}=k)$ and $I(\cdot)$ is an indicator function. The magnitude of θ measures the degree of the within-cluster correlation. We are interested in testing $H_0:\theta=0$

vs. $H_1 : \theta > 0$, where $H_0 : \theta = 0$ corresponds to no within-cluster correlation. Since the null hypothesis is on the boundary of the parameter space, neither the likelihood ratio test nor the Wald test follows a chi-square distribution asymptotically [38.40].

Li and *Lin* [38.33] considered a score test for H_0 and showed that it still follows a chi-square distribution asymptotically. Specifically, they showed that the score statistic of θ evaluated under $H_0 : \theta = 0$ is

$$U_\theta(\boldsymbol{\beta}) = \left.\frac{\partial \ell(\boldsymbol{\beta}, \theta)}{\partial \theta}\right|_{\theta=0} = \sum_{i=1}^{m} \frac{1}{2}\left[\frac{\partial^2 \ell_i(\boldsymbol{\beta}, b_i)}{\partial b_i^2} + \left(\frac{\partial \ell_i(\boldsymbol{\beta}, b_i)}{\partial b_i}\right)^2\right]\Bigg|_{b_i=0} \tag{38.27}$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left\{\left[\sum_{j=1}^{n_i}(\widetilde{Y}_{ij} - \widetilde{P}_{ij})\right]^2 - \sum_{j=1}^{n_i}\widetilde{P}_{ij}(1-\widetilde{P}_{ij})\right\}, \tag{38.28}$$

where

$$\widetilde{Y}_{ij} = \sum_{k=1}^{N-1} y_{ij,k} = I(Y_{ij} \le N-1)$$

and

$$\widetilde{P}_{ij} = \sum_{k=1}^{N-1} \exp(X'_{ij,k}\,\boldsymbol{\beta}) \Big/ \left[1 + \sum_{k=1}^{N-1} \exp(X'_{ij,k}\boldsymbol{\beta})\right]$$

is the mean of $\widetilde{Y}_{ij}$ under H_0. It is interesting to note that the form of (38.28) resembles the variance component score statistic for clustered binary data [38.31]. It can be shown that under $H_0 : \theta = 0$, $E[U_\theta(\boldsymbol{\beta})] = 0$ and $m^{-1/2}U_\theta(\boldsymbol{\beta})$ is asymptotically normal $MVN(0, I_{\theta\theta})$, where $I_{\theta\theta}$ is given in (38.30).

To study the properties of $U_\theta(\boldsymbol{\beta})$ under $H_1 : \theta > 0$, they expanded $E(\widetilde{Y}_{ij}|b_i)$ as a quadratic function of b_i, and showed that, under $H_1 : \theta > 0$,

$$E[U_\theta(\boldsymbol{\beta})] \approx \frac{1}{2}\sum_{i=1}^{m}\left[\sum_{j=1}^{n_i}\sum_{k\neq j}^{n_i} a_{ij}a_{ik} + \frac{1}{2}\sum_{j=1}^{n_i} a_{ij}\{a'_{ij}\}^2\right]\theta\,,$$

where $a_{ij} = \widetilde{P}_{ij}(1-\widetilde{P}_{ij})$ and $a'_{ij} = 1 - 2\widetilde{P}_{ij}$. As a result, $E[U_\theta(\boldsymbol{\beta})]$ is an increasing function of θ. Hence the test is consistent and one would expect a large value of $U_\theta(\boldsymbol{\beta})$ for a large value of θ.

Since $\boldsymbol{\beta}$ is unknown under H_0 and needs to be estimated, the score statistic for testing H_0 is

$$S = U_\theta(\hat{\boldsymbol{\beta}})/\widetilde{I}_{\theta\theta}^{1/2}(\hat{\boldsymbol{\beta}})\,, \tag{38.29}$$

where $\hat{\boldsymbol{\beta}}$ is the MLE of $\boldsymbol{\beta}$ under H_0 and can be easily obtained by fitting the generalized logistic model $\log(P_{ij,k}/P_{ij,N}) = \boldsymbol{X}'_{ij,k}\boldsymbol{\beta}$, (using SAS PROC CATMOD for example), and $\widetilde{I}_{\theta\theta} = I_{\theta\theta} - I_{\theta\boldsymbol{\beta}'}I^{-1}_{\boldsymbol{\beta}\boldsymbol{\beta}'}I_{\boldsymbol{\beta}\theta}$ is the efficient information of θ evaluated under $H_0 : \theta = 0$. Using L'Hôpital's rule, some calculations show that

$$I_{\theta\theta} = E\left[\left(\frac{\partial \ell}{\partial \theta}\right)^2\right] = \frac{1}{4}\sum_{i=1}^{m}\left[\sum_{j=1}^{n_i}\widetilde{P}_{ij}\widetilde{Q}_{ij}(1 - 6\widetilde{P}_{ij}\widetilde{Q}_{ij}) + 2\left(\sum_{j=1}^{n_i}\widetilde{P}_{ij}\widetilde{Q}_{ij}\right)^2\right], \tag{38.30}$$

$$I_{\boldsymbol{\beta}\boldsymbol{\beta}'} = \sum_{i=1}^{m} E\left(\frac{\partial \ell_i}{\partial \boldsymbol{\beta}}\frac{\partial \ell_i}{\partial \boldsymbol{\beta}'}\right) = \sum_{i=1}^{m}\mathbf{X}'_i\boldsymbol{\Sigma}_i\mathbf{X}_i\,, \tag{38.31}$$

$$I_{\theta\boldsymbol{\beta}'} = \sum_{i=1}^{m} E\left(\frac{\partial \ell_i}{\partial \theta}\frac{\partial \ell_i}{\partial \boldsymbol{\beta}'}\right) = \frac{1}{2}\sum_{i=1}^{m}\mathbf{P}'_i\{\mathbf{I}_{N-1}\otimes\mathbf{G}_i\}\mathbf{X}_i\,, \tag{38.32}$$

where the expectations are taken under H_0; $\mathbf{I}_{N-1}$ denotes an $(N-1)\times(N-1)$ identity matrix, and $\boldsymbol{X}_i = (\mathbf{X}'_{i1}, \ldots, \mathbf{X}'_{in_i})'$, where $\boldsymbol{X}_{ij} = (\boldsymbol{X}_{ij,1}, \ldots, \boldsymbol{X}_{ij,N-1})'$, $\tilde{Q}_{ij} = 1 - \tilde{P}_{ij}$, and $\boldsymbol{\Sigma}_i = (\boldsymbol{\Sigma}_{i,rl})$, which is an $(N-1)\times(N-1)$ block matrix whose (r,l)-th block is

$$\boldsymbol{\Sigma}_{i,rr} = \mathrm{diag}[P_{i1,r}(1-P_{i1,r}), \ldots, P_{in_i,r}(1-P_{in_i,r})]$$
$$\boldsymbol{\Sigma}_{i,rl} = \mathrm{diag}[-P_{i1,r}P_{i1,l}, \ldots, -P_{in_i,r}P_{in_i,l}]\,,\ r \neq l\,,$$

$\mathbf{G}_i = \mathrm{diag}(2\tilde{P}_{ij}^2 - 3\tilde{P}_{ij} + 1, \ldots, 2\tilde{P}_{in_i}^2 - 3\tilde{P}_{in_i} + 1)$ and $\mathbf{P}_i = (\boldsymbol{P}'_{i,1}, \ldots, \boldsymbol{P}'_{i,N-1})'$, where $\boldsymbol{P}_{i,r} = (P_{ij,r}, \ldots, P_{in_i,r})'$. Standard asymptotic calculations show that S is asymptotically $N(0,1)$ under H_0 and one rejects H_0 if S is large and the test is one-sided.

The score test S for $H_0 : \theta = 0$ has several attractive features. First, it can be easily obtained by fitting the generalized logistic model $\log(P_{ij,k}/P_{ij,N}) = \boldsymbol{X}'_{ij,k}\boldsymbol{\beta}$, which is model (38.25) under H_0, using standard software, such as SAS PROC CATMOD. Hence calculations of S do not involve any numerical integration. Secondly, it is the most powerful test locally. Finally it is robust, as no distribution is assumed for the random effect b_i. We discuss an application of the test based on (38.25) in Sect. 38.5.4.

38.5.2 The Variance Component Score Test in Random Effects Cumulative Probability Models

A widely used model for clustered ordinal data is the cumulative probability random effects model obtained by modeling the cumulative probabilities $r_{ij,k} = P(Y_{ij} \le k)$ as

$$g(r_{ij,k}) = \alpha_k + \boldsymbol{X}'_{ij}\boldsymbol{\beta}_x + b_i = \boldsymbol{X}'_{ij,k}\boldsymbol{\beta} + b_i \,, \quad k = 1, \ldots, N-1 \,, \tag{38.33}$$

where $g(\cdot)$ is a link function, $\boldsymbol{X}_{ij,k} = (\boldsymbol{e}'_k, \boldsymbol{X}'_{ij})'$, $\boldsymbol{\beta} = (\alpha_1, \cdots, \alpha_{N-1}, \boldsymbol{\beta}'_x)$, and $b_i \sim F(., \theta)$ for some distribution function F with zero mean and variance θ. For $g(\cdot) = \text{logit}(\cdot)$ and $g(\cdot) = \log[-\log(1-\cdot)]$, we have proportional odds and complementary log-log models. Define $o_{ij,k} = I(Y_{ij} \le k)$. Denote $\boldsymbol{r}_{ij} = (r_{ij,1}, \ldots, r_{ij,N-1})'$, $\boldsymbol{R}_i = (\boldsymbol{r}'_{i1}, \ldots, \boldsymbol{r}'_{in_i})'$ and define $\boldsymbol{o}_{ij}$, $\boldsymbol{O}_i$ similarly. Some calculations show that the score statistic of θ under $H_0 : \theta = 0$ is

$$U_\theta(\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{m}\Big[(\boldsymbol{O}_i - \boldsymbol{R}_i)'\boldsymbol{\Gamma}_i^{-1}\boldsymbol{H}_i b f 1_i b f 1'_i \boldsymbol{H}_i \boldsymbol{\Gamma}_i^{-1} \times (\boldsymbol{O}_i - \boldsymbol{R}_i) - b f 1'_i \tilde{\boldsymbol{W}}_i b f 1_i\Big] \,, \tag{38.34}$$

where $\boldsymbol{1}_i$ is an $n_i(N-1)\times 1$ vector of ones; the weight matrices of $\boldsymbol{H}_i$, $\boldsymbol{\Gamma}_i$ and $\tilde{\boldsymbol{W}}_i$ are given in Appendix A.2 of *Li* and *Lin* [38.33]. Though seemingly complicated, (38.34) essentially compares the empirical variance of the weighted responses to its nominal variance.

The score statistic for testing $H_0 : \theta = 0$ is $S = U_\theta(\hat{\boldsymbol{\beta}}) \,/\, \tilde{I}^{1/2}_{\theta\theta}(\hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is the MLE of $\boldsymbol{\beta}$ under H_0, and it can be easily obtained by fitting the standard cumulative probability model $g(r_{ij,k}) = \boldsymbol{X}'_{ij,k}\boldsymbol{\beta}$, and $\tilde{I}_{\theta\theta}(\hat{\boldsymbol{\beta}})$ is the efficient information of θ. Computing the information matrices is tedious since the calculations involve the third and fourth cumulants of a multinomial distribution. The explicit expressions of the information matrices are given in *Li* and *Lin* [38.33].

Standard asymptotic calculations show that the score statistic S follows $N(0,1)$ asymptotically below H_0, and has the same optimality and robustness properties stated at the end of Sect. 38.5.1. It can be easily calculated by fitting the standard cumulative probability model $g(r_{ij,k}) = \boldsymbol{X}'_{ij,k}\boldsymbol{\beta}$ using existing software, such as SAS PROC CATMOD, and does not require any numerical integration. Again a one-sided test is used and H_0 is rejected for a large value of S. An application of score test based on (38.33) is presented in Sect. 38.5.4.

38.5.3 Variance Component Tests in the Presence of Measurement Errors in Covariates

Li and *Lin* [38.33] extended the variance component score tests to the situation when covariates are measured with error. To proceed, we denote a vector of unobserved covariates (such as the true precipitation level or the true CD4 count) by $\boldsymbol{X}_{ij}$ and $\boldsymbol{C}_{ij}$ denotes other accurately measured covariates (such as rainfall location or patients' gender).

The random effects cumulative probability model (38.33) and the random effects generalized logistic model (38.25) can be written in a unified form

$$g(p_{ij,k}) = \alpha_k + \boldsymbol{X}'_{ij}\boldsymbol{\beta}_{x,k} + \boldsymbol{C}'_{ij}\boldsymbol{\beta}_{c,k} + b_i \,, \tag{38.35}$$

where b_i follows some distribution $F(., \theta)$ with mean 0 and variance θ. For the random effects cumulative probability model (38.33), $p_{ij,k} = r_{ij,k}$ and $\boldsymbol{\beta}_{x,1} = \ldots = \boldsymbol{\beta}_{x,N-1}$ and $\boldsymbol{\beta}_{c,1} = \ldots = \boldsymbol{\beta}_{c,N-1}$. For the random effects generalized logistic model (38.25), $p_{ij,k} = P_{ij,k}/P_{ij,N}$ and $g(\cdot) = \log(\cdot)$.

Suppose the observed covariates $\mathbf{W}_{ij}$ (such as radar measurements of rainfall or observed CD4 counts) measure $\boldsymbol{X}_{ij}$ (such as the true precipitation amount or the true CD4 counts) with error. It is customary to postulate a nondifferential additive measurement error model for $\mathbf{W}_{ij}$ [38.41],

$$\mathbf{W}_{ij} = \mathbf{X}_{ij} + \mathbf{U}_{ij} \,, \tag{38.36}$$

where the $\mathbf{U}_{ij}$ are independent measurement errors following $MVN(0, \boldsymbol{\Sigma}_u)$. Suppose that the measurement error covariance $\boldsymbol{\Sigma}_u$ is known or is estimated as $\hat{\boldsymbol{\Sigma}}_u$, using replicates or validation data for example. We are interested in testing for no within-cluster correlation $H_0 : \theta = 0$ in the random effects measurement error models (38.35) and (38.36). *Li* and *Lin* [38.33] have proposed using the SIMEX method by extending the results in the previous two sections to construct score tests for H_0 to account for measurement errors.

Simulation extrapolation (SIMEX) is a simulation-based functional method for inference on model parameters in measurement error problems [38.36], where no distributional assumption is made about the unobserved covariates $\mathbf{X}_{ij}$. We first briefly describe parameter estimation in random effects measurement error models (38.35, 36) using the SIMEX method, then discuss how to use the SIMEX idea to develop SIMEX score tests for $H_0 : \theta = 0$.

The SIMEX method involves two steps: the simulation step and the extrapolation step. In the simulation step, data $\mathbf{W}_{ij}^*$ is generated by adding to $\mathbf{W}_{ij}$ a random error following $N(0, \eta\boldsymbol{\Sigma}_u)$ for some constant $\eta > 0$. Naive parameter estimates are then calculated by fitting (38.35) with $\boldsymbol{X}_{ij}$ replaced by $\mathbf{W}_{ij}^*$. This gives the naive estimates if the measurement error covariance is $(1+\eta)\boldsymbol{\Sigma}_u$. This procedure is repeated for a large number B of times (for example $B = 100$), and the mean of the resulting B naive parameter estimates is calculated. One does this for a series of values of η (such as $\eta = 0.5, 1, 1.5, 2$). In the extrapolation step, a regression (such as a quadratic) model is fitted to the means of these naive estimates as a function of η, and is extrapolated to $\eta = -1$, which corresponds to zero measurement error variance. These extrapolated estimates give the SIMEX estimates for the model parameters. For details of the SIMEX method, see *Cook* and *Stefanski* [38.36] and *Carroll* et al. [38.41]. The SIMEX idea can be utilized to construct score tests for $H_0 : \theta = 0$ in the random effects measurement error models (38.35) and (38.36) by extending the results in Sects. 38.5.1 and 38.5.2. The resulting SIMEX score tests are an extension of the work of *Lin* and *Carroll* [38.37] to random effects measurement error models for clustered polytomous data.

In the absence of measurement error, the score statistics for testing $H_0 : \theta = 0$ under (38.35) take the same form $U_\theta(\hat{\boldsymbol{\beta}}) \,/\, \widetilde{I}_{\theta\theta}^{1/2}(\hat{\boldsymbol{\beta}})$, where $U_\theta(\hat{\boldsymbol{\beta}})$ is given in (38.34) for random effects cumulative probability models and in (38.28) for random effects generalized logistic models. The denominator $\widetilde{I}_{\theta\theta}(\hat{\boldsymbol{\beta}})$ is in fact the variance of $U_\theta(\hat{\boldsymbol{\beta}}|)$. The main idea of the SIMEX variance component score test is to treat the score statistic in the numerator $U_\theta(\cdot)$ as if it were a parameter estimator and use the SIMEX variance method (*Carroll* et al. [38.41]) to calculate the variance of this 'estimator'. Specifically, in the SIMEX simulation step, one simply calculates naive score statistics using the score formulae (38.34) and (38.28) by replacing $\boldsymbol{X}_{ij}$ with the simulated data $\boldsymbol{W}_{ij}^*$. The rest of the steps parallel those in the standard SIMEX method for parameter estimation. Denoting the results by $U_{\text{simex}}(\cdot)$ and $\widetilde{I}_{\theta\theta,\text{simex}}$ respectively, the SIMEX score statistic is simply

$$S_{\text{simex}} = U_{\text{simex}} / \widetilde{I}_{\theta\theta,\text{simex}}^{1/2} , \qquad (38.37)$$

which follows $N(0, 1)$ asymptotically when the true extrapolation function is used. Since the true extrapolation function is unknown in practice, an approximation (such as a quadratic) is used. The simulation study reported by *Li* and *Lin* [38.33] shows that the SIMEX score tests perform well. The theoretical justification for the SIMEX score tests can be found in *Lin* and *Carroll* [38.37].

The SIMEX score test possesses several important advantages. First, it can be easily calculated by fitting standard cumulative probability models using available software such as SAS PROC CATMOD. Secondly, it is robust in the sense that no distribution needs to be assumed for the frailty b_i and for the unobserved covariates $\boldsymbol{X}$.

38.5.4 Data Examples

To illustrate the variance component score tests for clustered polytomous data, we examine data from a longitudinal study on the efficacy of steam inhalation for treating common cold symptoms, conducted by *Macknin* et al. [38.42]. This study included 30 patients with colds of recent onset. At the time of enrolment, each patient went through two 20 min steam inhalation treatments spaced 60–90 minutes apart. Assessment of subjective response was made on an individual daily score card by the patient from day 1 (baseline) to day 4. On each day, the severity of nasal drainage was calibrated into four ordered categories (no symptoms, mild, moderate and severe symptoms). The study examined whether the severity improved following the treatment, and tested whether the observations over time for each subject were likely to be correlated.

Li and *Lin* [38.33] considered models (38.25) and (38.33) with the time from the baseline as a covariate. They first assumed a random effects logistic model (38.25), and obtained a variance component score statistic 5.32 (p-value < 0.001), which provided strong evidence for within-subject correlation over time. Similar results were found when they fitted a random effects proportional odds model (38.33) (score statistic = 9.70, p-value < 0.001). In these two tests they assumed no distribution for the random effect b_i.

To further examine the effect of time, they fitted (38.33) by further assuming that the random effect b_i followed $N(0, \theta)$. The MLE of the coefficient of time was -0.33 (SE $= 0.21$), which suggested that the severity improved following the treatment but that improvement was not statistically significant (p-value $= 0.11$). The estimated variance component was 2.31 (SE $= 0.45$). This result was consistent with the test results.

38.6 Discussion

Central to the idea of mixed modeling is the idea of fixed and random effects. Each effect in a model must be classified as either a fixed or a random effect. Fixed effects arise when the levels of an effect constitute the entire population of interest. For example, if an industrial experiment focused on the effectiveness of three brands of a machine, *machine* would be a fixed effect only if the experimenter's interest did not go beyond the three machine brands. On the other hand, an effect is classified as a random effect when one wishes to make inferences on an entire population, and the levels in the experiment represent only a sample from that population. Consider an example of psychologists comparing test results between different groups of subjects. Depending on the psychologists' particular interest, the group effect might be either fixed or random. For example, if the groups are based on the sex of the subject, *sex* would be a fixed effect. But if the psychologists are interested in the variability in test scores due to different teachers, they might choose a random sample of teachers as being representative of the total population of teachers, and *teacher* would be a random effect. Returning to the machine example presented earlier, *machine* would also be considered to be a random effect if the scientists were interested in making inferences on the entire population of machines and randomly chose three brands of machines for testing.

In summary, what makes a random effect unique is that each level of a random effect contributes an amount that is viewed as a sample from a population of random variables. The estimate of the variance associated with the random effect is known as the variance component because it measures the part of the overall variance contributed by that effect. In mixed models, we combine inferences about means (of fixed effects) with inferences about variances (of random effects).

A few difficulties arise from setting up the likelihood function to draw inference based on a random effects model. The major obstacle lies in computation, as, for practitioners, the main issue focuses on how to handle the intractable MLE calculations. This chapter reviews some commonly used approaches to estimating the regression coefficients and the variance components in the (generalized) linear mixed models. We note that the EM algorithm can yield maximum likelihood estimates, which are consistent and most efficient under regularity conditions. However, its computational burden is substantial, and the convergence rate is often slow. Laplace approximation greatly reduces the computational load, but the resulting estimates are generally biased. The simulated maximum likelihood estimation is considerably less computationally burdensome compared to the EM. For example, the rejection sampling is avoided, saving much computing time. However, its obvious drawback is the local convergence – a 'good' initial point is required to achieve the global maximizer. The so-called SA and S–U algorithms seem to be promising, as they make full use of the simulated data and obtain the estimates recursively. However, the detailed implementation of both methods have yet to be finalized in the literature.

It is worth briefly discussing marginal models, another major tool for handling clustered data. In a marginal model, the marginal mean of the response vector is modeled as a function of explanatory variables [38.43]. Thus, in contrast to the random effect models, the coefficients in a marginal model have population average interpretations. This type of model is typically fitted via the so-called generalized estimating equation (GEE). An appealing feature is that, for the right mean structure, even when the covariance structure of the response is mis-specified, the GEE acquires consistent estimates. However, the GEE method faces several difficulties, which may easily be neglected. First, the GEE estimator's efficiency becomes problematic when the variance function is mis-specified. Secondly, the consistency of the estimator is only guaranteed under noninformative censoring; informative censoring generally leads to biased estimates. More related discussion can be found in *Diggle* et al. [38.43].

Lastly, we point out other active research areas in mixed modeling, including evaluating the model's goodness of fit, choosing the best distribution for the random effects and selecting the best collection of covariates for a model. Readers are referred to some recent articles on these topics (such as [38.44–47]).

References

38.1 J. A. Nelder, R. W. Wedderburn: Generalized linear models, J. R. Stat. Soc. A **135**, 370–384 (1972)

38.2 P. McCullagh, J. A. Nelder: *Generalized Linear Models*, 2 edn. (Chapman Hall, London 1989) 1st edition, 1983

38.3 N. M. Laird, J. H. Ware: Random-effects models for longitudinal data, Biometrics **38**, 963–974 (1982)

38.4 R. Stiratelli, N. M. Laird, J. H. Ware: Random effects models for serial observations with binary response, Biometrics **40**, 961–971 (1984)

38.5 R. Schall: Estimation in generalized linear models with random effects, Biometrika **78**, 719–727 (1991)

38.6 S. L. Zeger, M. R. Karim: Generalized linear model with random effects: a Gibbs sampling approach, J. Am. Stat. Assoc. **86**, 79–86 (1991)

38.7 C. E. McCulloch: Maximum likelihood algorithms for generalized linear mixed models, J. Am. Stat. Assoc. **92**, 162–170 (1997)

38.8 N. E. Breslow, D. G. Clayton: Approximate inference in generalized linear mixed models, J. Am. Stat. Assoc. **88**, 9–25 (1993)

38.9 N. A. Cressie: *Statistics for Spatial Data* (Wiley, New York 1991)

38.10 D. A. Harville: Bayesian inference for variance components using only error contrasts, Biometrika **61**, 383–385 (1974)

38.11 C. E. McCulloch, S. R. Searle: *Generalized, Linear, and Mixed Models* (Wiley, New York 2001)

38.12 C. R. Henderson, O. Kempthorne, S. R. Searle, C. N. von Krosigk: Estimation of environmental, genetic trends from records subject to culling, Biometrics **15**, 192–218 (1959)

38.13 C. Bliss: The method of probits, Science **79**, 38–39 (1934)

38.14 P. J. Diggle, J. A. Tawn, R. A. Moyeed: Model-based geostatistics, J. R. Stat. Soc. C-AP **47**, 299–326 (1998)

38.15 A. P. Dempster, N. M. Laird, D. B. Rubin: Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B **39**, 1–22 (1977)

38.16 N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth: Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087–1092 (1953)

38.17 W. Hastings: Monte Carlo sampling methods using Markov chains and their applications, Biometrika **57**, 97–109 (1970)

38.18 B. P. Carlin, T. A. Louis: *Bayes and Empirical Bayes Methods for Data Analysis* (Chapman Hall, New York 2000)

38.19 A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rubin: *Bayesian Data Analysis* (Chapman Hall, London 1995)

38.20 C. J. Geyer, E. A. Thompson: Constrained Monte Carlo maximization likelihood for dependent data, J. R. Stat. Soc. B **54**, 657–699 (1992)

38.21 A. E. Gelfand, B. P. Carlin: Maximum likelihood estimation for constrained- or missing-data problems, Can. J. Stat. **21**, 303–311 (1993)

38.22 C. P. Robert, G. Casella: *Monte Carlo Statistical Methods* (Springer, Berlin Heidelberg 1999)

38.23 M. A. Tanner, W. H. Wong: The calculation of posterior distributions by data augmentation, J. Am. Stat. Assoc. **82**, 528–549 (1987)

38.24 G. Satten, S. Datta: The S-U algorithm for missing data problems, Comp. Stat. **15**, 243–277 (2000)

38.25 G. Satten: Rank-based inference in the proportional hazards model for interval censored data, Biometrika **83**, 355–370 (1996)

38.26 P. J. Green: Penalized likelihood for general semi-parametric regression models, Int. Stat. Rev. **55**, 245–259 (1987)

38.27 Y. Lee, J. A. Nelder: Hierarchical generalized linear models, J. R. Stat. Soc. B **58**, 619–678 (1996)

38.28 Q. Liu, D. A. Pierce: Heterogeneity in Mantel-Haenszel-type models, Biometrika **80**, 543–556 (1993)

38.29 P. J. Solomon, D. R. Cox: Nonlinear component of variance models, Biometrika **79**, 1–11 (1992)

38.30 X. Lin, N. E. Breslow: Bias correction in generalized linear mixed models with multiple components of dispersion, J. Am. Stat. Assoc. **91**, 1007–1016 (1996)

38.31 D. Commenges, L. Letenneur, H. Jacqmin, J. Moreau, J. Dartigues: Test of homogeneity of binary data with explanatory variables, Biometrics **50**, 613–20 (1994)

38.32 X. Lin: Variance component testing in generalized linear models with random effects, Biometrika **84**, 309–326 (1997)

38.33 Y. Li, X. Lin: Testing random effects in uncensored/censored clustered data with categorical responses, Biometrics **59**, 25–35 (2003)

38.34 C. G. Collier: *Applications of Weather Radar Systems: A Guide to Uses of Radar in Meteorology and Hydrology* (Wiley, New York 1996)

38.35 A. A. Tsiatis, V. Degruttola, M. S. Wulfsohn: Modeling the relationship of survival to longitudinal data measured with error: applications to survival, CD4 counts in patients with AIDS, J. Am. Stat. Assoc. **90**, 27–37 (1995)

38.36 J. R. Cook, L. A. Stefanski: Simulation-extrapolation estimation in parametric measurement error models, J. Am. Stat. Assoc. **89**, 1314–1328 (1994)

38.37 X. Lin, R. J. Carroll: SIMEX variance component tests in generalized linear mixed measurement error models, Biometrics **55**, 613–619 (1999)

38.38 D. A. Harville, R. W. Mee: A mixed-model procedure for analyzing ordered categorical data, Biometrics **40**, 393–408 (1984)

38.39 D. Hedeker, R. Gibbons: A random-effects ordinal regression model for multilevel analysis, Biometrics **50**, 933–945 (1994)

38.40 S. G. Self, K. Y. Liang: Asymptotic properties of maximum likelihood estimators, likelihood ratio tests under nonstandard conditions, J. Am. Stat. Assoc. **82**, 605–610 (1987)

38.41 R. J. Carroll, D. Ruppert, L. A. Stefanski: *Measurement Error in Nonlinear Models* (Chapman Hall, London 1995)

38.42 M. L. Macknin, S. Mathew, S. V. Medendorp: Effect of inhaling heated vapor on symptoms of the

common cold, J. Am. Med. Assoc. **264**, 989–991 (1990)
38.43 P.J. Diggle, K.Y. Liang, S.L. Zeger: *Analysis of longitudinal data* (Oxford Univ. Press, New York 1994)
38.44 B. Zheng: Summarizing the goodness of fit of generalized linear models for longitudinal data, Stat. Med. **19**, 1265–1275 (2000)
38.45 G. Verbeke, E. Lesaffre: A linear mixed-effects model with heterogeneity in the random-effects population, J. Am. Stat. Assoc. **91**, 217–221 (1996)
38.46 P.J. Lindsey, J.K. Lindsey: Diagnostic tools for random effects in the repeated measures growth curve model, Comput. Stat. Data Anal. **33**, 79–100 (2000)
38.47 E.A. Houseman, L.M. Ryan, B.A. Coull: Cholesky residuals for assessing normal errors in a linear model with correlated outcomes, J. Am. Stat. Assoc. **99**, 383–394 (2004)

39. Cluster Randomized Trials: Design and Analysis

The first section of this chapter gives an introduction to cluster randomized trials, and the reasons why such trials are often chosen above simple randomized trials. It also argues that more advanced statistical methods for data obtained from such trials are required, since these data are correlated due to the nesting of persons within clusters. Traditional statistical techniques, such as the regression model ignore this dependency, and thereby result in incorrect conclusions with respect to the effect of treatment. In the first section it is also argued that the design of cluster randomized trials is more complicated than that of simple randomized trials; not only the total sample size needs to be determined, but also the number of clusters and the number of persons per cluster.

The second section describes and compares the multilevel regression model and the mixed effects analysis of variance (ANOVA) model. These models explicitly take into account the nesting of persons within clusters, and thereby the dependency of outcomes of persons within the same cluster. It is shown that the traditional regression model leads to an inflated type I error rate for treatment testing.

Optimal sample sizes for cluster randomized trials are given in Sects. 39.3 and 39.4. These sample sizes can be shown to depend on the intra-class correlation coefficient, which measures the amount of variance in the outcome variable at the cluster level. A guess of the true value of this parameter must be available in the design stage in order to calculate the optimal sample sizes. Section 39.5 focuses on the robustness of the optimal sample size against incorrect guesses of this parameter. Section 39.6 focuses on optimal designs when the aim is to estimate the intra-class correlation with the greatest precision.

Cluster randomized trials are experiments in which complete clusters of persons, rather than the persons themselves, are randomized to treatment conditions. Such trials are frequently used in the agricultural, (bio-)medical, social, and behavioral sciences. Examples are school-based smoking prevention and cessation interventions with pupils nested within classes within schools, clinical trials with patients nested within clinics or general practices, and studies on interventions to reduce absences due to sick leave with employers nested within divisions within companies. Cluster randomized trials are very natural in the case of existing clusters, but can also be used when groups are created for the purpose of the trial. An example is a trial with therapy groups to reduce alcohol addiction. Alcoholics are assigned to small therapy groups, which in turn are assigned to treatment conditions. The difference is that, in trials with existing clusters, the persons also meet and interact outside the time slots during which the intervention is delivered, resulting in a larger degree of mutual influence.

39.1 Cluster Randomized Trials

Cluster randomized trials are often chosen above simple randomized trials that randomize persons to treatment conditions, although cluster randomized trials can easily be shown to be less efficient (Sect. 39.2). The reasons why cluster randomized trials are so often adopted must rest on other considerations than statistical efficiency, and these are often of an administrative, financial, political or ethical nature. As an example consider a study on the impact of vitamin A supplementation on childhood mortality [39.1]. In this study complete villages in Indonesia were randomly assigned to either the supplementation or control group because it was not considered politically feasible to randomize children. Another advantage of adopting a cluster randomized trial in this example is that the capsules containing the vitamin A supplements only have to be delivered to those villages in the supplementation group, which results in a reduction of travel costs. A trial that randomizes children would suffer from control-group contamination if the children in the control group were able to get access to the vitamin A from children in the supplementation group in the same village. In some cases there is no alternative to cluster randomized trials, such as in community-based interventions where the intervention will necessarily affect all members in the community. Another reason to adopt a cluster randomized trial is the wish to increase compliance. It may be reasonable to expect that compliance increases in a study where complete families, rather that just a few family members, are randomized to treatment conditions.

Due to the nesting of persons within clusters, the design and analysis of cluster randomized trials is more complicated than for simple randomized trials. The traditional assumption of independence is by definition violated when data have a nested structure. This is obvious, since there is mutual influence among persons within the same cluster. So a person's opinion, behavior, attitude or health is influenced by that of other persons in the same cluster. Furthermore, persons are influenced by cluster policy and cluster leaders. In school-based smoking prevention interventions, for instance, a pupil's smoking behavior is influenced by that of other pupils within the same class and (to a lesser degree) school, that of teachers, the school policy towards smoking and the availability of cigarettes and advertisements on smoking in the school and its neighborhood.

The traditional regression model, which assumes independent outcomes, cannot be used for the analysis of nested data. Naively using this model may lead to incorrect point estimates and standard errors of regression coefficients, and therefore to incorrect conclusions on the effect of treatment conditions and covariates on the outcome [39.2–5]. The appropriate model is the multilevel model [39.6], which is also referred to as the hierarchical (linear) model [39.7], or random coefficient model [39.6]. The multilevel model treats the persons as the unit of analysis, but explicitly takes into account nesting of persons within clusters and the correlation of outcomes of persons within the same cluster. It assumes that the clusters in the study represent a random sample from their population, and treats their effects as random in the regression analysis so that the results can be generalized to this population. Multilevel models are an extension of the variance components models and mixed effects ANOVA (analysis of variance) models [39.8] that have long been used in the biological and agricultural sciences. They are an extension in the sense that they do not only include categorical, but also continuous explanatory variables. They have been developed since the early 1980s, and in the first instance especially gained attention from the educational sciences, where data by nature have a multilevel structure due to the nesting of pupils within classes within schools. Nowadays, multilevel models are used in various fields of science, ranging from political sciences to nursery, and from studies on interviewer effects to studies with longitudinal data. It is to be expected that multilevel analysis will become part of the standard statistical techniques in the near future and that editors of scientific journals will no longer consider contributions that use old-fashioned methods to analyze multilevel data.

The design of cluster randomized trials is more complicated than that of simple randomized trials, since it does not only involve the calculation of the required number of persons, but also the calculation of the optimal allocation of units, that is, the optimal sample sizes at the cluster level and the person level. One may wonder if it is more efficient to sample many small clusters or to sample just a few large clusters. Of course, the number of available clusters is limited and the optimal number of clusters cannot therefore be larger than the available number of clusters. Likewise, the optimal cluster size cannot be larger than the actual cluster size, and such preconditions must be taken into account when calculating the optimal design. Furthermore, it is often less expensive to sample a person within an already sampled cluster than to sample a new cluster. So, the costs of sampling persons and clusters and the available budget

should also be taken into account, and it is worthwhile to calculate the required budget to achieve a pre-specified power level to detect a relevant treatment effect, or vice versa, the maximum power level given a fixed budget.

A concern in the design of cluster randomized trials is the fact that the optimal design depends on the true value of the intra-class correlation coefficient, a parameter which measures the amount of variance of the outcome variable at the cluster level. Of course, the true value is not known at the design stage, and an educated guess of this parameter must be used instead. Such a guess can be obtained from knowledge of the subject matter or from similar studies in the past. There is, however, no guarantee that such an educated guess is correct, and it is therefore worthwhile to study the robustness of optimal designs against an incorrect prior value of the intra-class correlation coefficient, and to development robust optimal design techniques.

The contents of this chapter are as follows. In the next section the multilevel regression model and the mixed effects ANOVA model are described and compared. Section 39.3 gives formulae for the optimal allocation of units for models without covariates. The extension to models with covariates is the topic of Sect. 39.4. In Sect. 39.5 we focus on the robustness properties of optimal designs. In Sect. 39.6 we present designs that are useful when interest lies in the degree of the intra-class correlation. Finally, conclusions and a discussion are given in Sect. 39.7. For the sake of simplicity, we focus on optimal designs for models with two levels of nesting, two treatment conditions, and a continuous outcome.

39.2 Multilevel Regression Model and Mixed Effects ANOVA Model

In the simplest version of a cluster randomized trial we wish to compare the effects of an intervention and a control on a single continuous outcome variable. The multilevel regression model relates outcome y_{ij} for person i in cluster j to treatment condition z_j

$$y_{ij} = \beta_0 + \beta_1 z_j + u_j + e_{ij} \,. \tag{39.1}$$

In this chapter, the treatment condition is coded $z_j = -0.5$ for the control condition and $z_j = +0.5$ for the intervention condition. So, β_0 is the mean outcome, and β_1 is the treatment effect, which is estimated by the difference in mean outcomes in both treatment groups. The null hypothesis of no treatment effect is tested by the statistic $t = \hat{\beta}_1 / \sqrt{\hat{\text{var}}(\hat{\beta}_1)}$, which has a t-distribution with $n_2 - 2$ degrees of freedom under the null hypothesis.

The multilevel model differs from the traditional regression model since it contains two random error terms. The term $u_j \sim N(0, \tau^2)$ at the cluster level is the deviation of cluster j from the mean outcome in its treatment condition, and the term $e_{ij} \sim N(0, \sigma^2)$ at the person level is the deviation of person i from the mean outcome in cluster j. These two error terms are assumed to be independent of each other and of possible covariates in the model. In general, the within-cluster variance σ^2 is much larger than the between-cluster variance τ^2.

The first two terms and the right-hand side of (39.1) are the fixed part of the model, whereas the second two terms are the random part. Since it contains both fixed and random effects, the multilevel model is a mixed effects model. Fixed effects are effects that are attributable to a finite set of levels of a factor, and they occur in the data because interest lies only in them, and not in any other levels of that factor. An example of a fixed effect is a treatment factor in a smoking prevention intervention with two levels: intervention and control. We are only interested in the comparison of these two treatment conditions, and not in any other. Random effects, on the other hand, are attributable to an infinite set of levels of a factor, of which only a random sample is included in the study at hand. An example of a random effect is the school effect in a school-based smoking prevention intervention. Although not all schools of the population under study are included in the study, we wish to generalize its findings to all schools in the population. Therefore, school is included as a random effect rather than a fixed effect.

The variances σ^2 and τ^2 are called the variance components since they sum up to the total variance of the outcome of person i within cluster j:

$$\text{var}(y_{ij}) = \sigma^2 + \tau^2 \,. \tag{39.2}$$

Furthermore, there is correlation between outcomes of two persons within the same cluster j:

$$\text{cov}(y_{ij}, y_{i'j}) = \tau^2 \,. \tag{39.3}$$

The intra-class correlation coefficient ρ measures the proportion of variation in the outcomes at the cluster

level, that is

$$\rho = \frac{\mathrm{var}(y_{ij})}{\mathrm{cov}(y_{ij}, y_{i'j})} = \frac{\tau^2}{\sigma^2 + \tau^2} \,. \tag{39.4}$$

This parameter may be interpreted as the standard Pearson correlation coefficient between any two outcomes in the same cluster. Intra-class correlation coefficients are often considerably larger in small clusters such as households, than in large clusters such as postcode levels. This can be explained by the fact that members in small clusters meet each other more often, which results in a higher level of mutual influence. As we will see in the next section, the intra-class correlation coefficient plays a crucial role in calculating the optimal sample sizes.

In a balanced design, randomization is done such that both treatment conditions have $\frac{1}{2}n_2$ clusters, and each cluster consists of n_1 persons. The variance of the treatment effect estimator is then given by

$$\mathrm{var}(\hat{\beta}_1) = 4\frac{\sigma^2 + n_1\tau^2}{n_1 n_2} = 4\frac{\sigma^2 + \tau^2}{n_1 n_2}\left[1 + (n_1 - 1)\rho\right] \,. \tag{39.5}$$

This variance is larger than that obtained with the traditional regression model due to the inclusion of the factor $[1 + (n_1 - 1)\rho]$. This factor is called the design effect, and it increases with the cluster size n_1 and the intra-class correlation coefficient ρ. Since it is always larger than 1, a cluster randomized trial is less efficient than a trial that randomizes persons to treatment conditions. Even for small values of ρ, the design effect may already be considerable. For example, if $\rho = 0.1$ and $n_1 = 10$ the design effect is equal to 1.9, and so the $\mathrm{var}(\hat{\beta}_1)$ as obtained with the multilevel model is about twice that obtained with ordinary regression analysis. So, incorrectly using the traditional regression model results in a value of $\mathrm{var}(\hat{\beta}_1)$ that is too low, and consequently in an inflated type I error rate [39.2]. This is especially the case when the cluster size n_1 and the intra-class correlation coefficient ρ are large.

When treatment condition is the only predictor variable we can write the multilevel model in terms of a mixed effects ANOVA model. For person $i = 1, \ldots, n_1$ in cluster $j = 1, \ldots, n_2$ in treatment $t = 1, 2$ we have

$$y_{ijt} = \mu + \alpha_t + u_{jt} + e_{ijt} \,. \tag{39.6}$$

Here, μ is the grand mean, α_t is the fixed effect of the t-th treatment, and u_{jt} and e_{ijt} are the random effects at the cluster and person level, which are assumed to be normally distributed with zero mean and variances of τ^2 and σ^2 respectively. Since clusters are nested within treatment conditions, we have a nested ANOVA model.

When $t = 1$ corresponds to the control group and $t = 2$ corresponds to the intervention group the correspondence between the parameters in the multilevel regression model in (39.1) and the mixed effects ANOVA model in (39.6) is given by

$$\mu = \beta_0, \quad \alpha_2 - \alpha_1 = \beta_1, \quad u_{jt} = u_j, \quad e_{ijt} = e_{ij} \,. \tag{39.7}$$

Table 39.1 gives the expected means squares (MS) for the mixed effect ANOVA model. The test statistic for the null hypothesis of no treatment effect is given by $F = MS_{\mathrm{treatment}}/MS_{\mathrm{cluster}}$, which, under the null hypothesis, has an F-distribution with 1 and $n_2 - 2$ degrees of freedom. The value of the F-test statistic for the mixed effects ANOVA model can be shown to be equal to the square of the value of the t-test statistic for the multilevel regression model [39.9]. The two variance components are estimated by

$$\hat{\sigma}^2 = \mathbf{MS}_{\mathrm{person}} \,, \tag{39.8}$$

and

$$\hat{\tau}^2 = \left(\mathbf{MS}_{\mathrm{cluster}} - \mathbf{MS}_{\mathrm{person}}\right)/n \,, \tag{39.9}$$

and the intra-class correlation coefficient is estimated by

$$\hat{\rho}^2 = \frac{\mathbf{MS}_{\mathrm{cluster}} - \mathbf{MS}_{\mathrm{person}}}{\mathbf{MS}_{\mathrm{cluster}} + (n-1)\mathbf{MS}_{\mathrm{person}}} \,. \tag{39.10}$$

For a long time the estimation of mixed models was a difficulty because of the lack of suitable estimation methods and computer programs. Different models

Table 39.1 Values for the mixed effects ANOVA model

Source	Degrees of freedom	Mean squares	Expected MS
Treatment	1	$MS_{\mathrm{treatment}}$	$\sigma^2 + n_1\tau^2 + n_1 n_2 \sum_t \alpha_t^2$
Clusters within treatment	$n_2 - 2$	MS_{cluster}	$\sigma^2 + n_1\tau^2$
Persons within clusters	$n_1 n_2 - n_2$	MS_{person}	σ^2
Total	$\boldsymbol{n_1 n_2 - 1}$		

were used but these can be shown to result in incorrect estimates of regression coefficients and their standard errors [39.2]. One such model is the traditional ordinary regression model, which assumes independent outcomes and thereby ignores nesting of persons within clusters and correlation of outcomes within the same cluster. Another approach is to calculate mean scores of variables at the cluster level and to use these in a regression model. With this approach, clusters are used as the unit of analysis, which results in loss of information. A third approach is to include clusters as fixed effects in the regression model, even if the results have to be generalized to the populations of clusters.

A method for estimation of mixed effects model became available with the development of full-information maximum-likelihood (ML), and restricted maximum-likelihood estimators (REML). The first calculates the regression coefficients and (co-)variance components such that the log likelihood log (L) is maximized, where

$$\log(L) = -\frac{1}{2}\sum_j n_{1j}\log 2\pi - \frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})\,. \qquad (39.11)$$

The vector $\boldsymbol{y}$ is the vector of outcomes, $\boldsymbol{\beta}$ is the vector of regression coefficients, and $\mathbf{V}$ is the covariance matrix of the outcomes, which is a function of the variance components. The design matrix $\mathbf{X}$ contains the measures on the predictor variables. REML is an adjustment of ML since it takes into account the loss of degrees of freedom resulting from estimating the fixed effects while estimating the variance components. So, the ML estimates of the variance components are downward-biased, while those for REML are not. For a large number of clusters (say $n_2 > 30$), the difference between the two estimates is negligible.

During the 1980s much attention was paid to the development of methods for the computation of ML and REML estimates, such as iterative generalized least squares (IGLS) [39.10], and restricted iterative generalized least squares (RIGLS) [39.11], which in the normal case produce ML and REML estimates, respectively. Furthermore, attention was paid to the application of existing methods, such as the Fisher scoring algorithm [39.12], and the expectation-maximization (EM) algorithm [39.13, 14]. The introduction and widespread use of personal computers have initiated the development of specialized computer programs for multilevel analysis, such as MLwin [39.15] and HLM [39.16]. Nowadays, multilevel analysis is part of general-purpose statistical software, such as SPSS and STATA.

39.3 Optimal Allocation of Units

39.3.1 Minimizing Costs to Achieve a Fixed Power Level

The primary aim of an experiment is to gain insight into the magnitude of the treatment effect, and to test if it is different from zero. Thus, we wish to test the null hypothesis $H_0 : \beta_1 = 0$ against the alternative that its value is different from zero. This hypothesis is tested by the test statistic $t = \hat{\beta}_1/\sqrt{\hat{\text{var}}(\hat{\beta}_1)}$, which has a t-distribution with $n_2 - 2$ degrees of freedom under the null hypothesis. When the number of clusters is large, the standard normal distribution can be used as an approximation, as will be done in the remainder of this chapter. For a two-sided alternative hypothesis $H_1 : \beta_1 \neq 0$, the power $1-\gamma$, type I error rate α, and the true value of β_1 are related to the variance $\text{var}(\hat{\beta}_1)$ as follows:

$$\text{var}(\hat{\beta}_1) = \left(\frac{\beta_1}{z_{1-\alpha/2} + z_{1-\gamma}}\right)^2, \qquad (39.12)$$

where $z_{1-\alpha/2}$ and $z_{1-\gamma}$ are the $100(1-\alpha/2)\%$ and $100(1-\gamma)\%$ standard normal deviates. For a one-sided alternative hypothesis, $1-\alpha/2$ may be replaced by $1-\alpha$. In general, the true value of the treatment effect β_1 is unknown at the design stage, and it is replaced by the minimal relevant deviation of β_1 from zero. If this effect is expressed in terms of units of the standard deviation $\sqrt{\sigma^2+\tau^2}$ of the outcome y_{ij}, then it is a relative treatment effect. Relative treatment effects equal to 0.2, 0.5, and 0.8 can be considered small, medium, and large, respectively, where a medium treatment effect is visible to the naked eye of a careful researcher [39.17].

As follows from (39.12), the power increases with the true value of β_1, which is obvious since large treatment effects are easier to detect than small treatment effects. Also, the power increases with the type I error rate, since null hypotheses are easier rejected if the probability of a type I error is large. Furthermore, the power is inversely related to the $\text{var}(\hat{\beta}_1)$. So, maximizing the power corresponds to minimizing the variance

of the estimated treatment effect. For studies with non-nested data this variance is related to the total sample size, and minimal sample sizes can be found in, for instance, *Cochran* [39.18]. For studies with two levels of nesting, $\text{var}(\hat{\beta}_1)$ does not only depend on the total sample size $n_1 n_2$, but also on the cluster size n_1, as follows from (39.5). Note that we use non-varying cluster sizes since that leads to the most efficient design [39.19]. In reality, cluster sizes generally vary, so that we have to take a sample of size n_1 from each cluster, meaning that not all persons in the sampled clusters are enrolled in the experiment.

The required sample sizes n_1 and n_2 can be calculated by substituting $\text{var}(\hat{\beta}_1)$ from (39.5) into (39.12). For fixed cluster size n_1 the required number of clusters is equal to

$$n_2 = 4\frac{\sigma^2+\tau^2}{n_1}\left[1+(n_1-1)\rho\right]\left(\frac{z_{1-\alpha/2}+z_{1-\gamma}}{\beta_1}\right)^2 . \tag{39.13}$$

For a fixed number of clusters n_2, the required cluster size is equal to

$$n_1 = \frac{4\sigma^2}{\left(\frac{\beta_1}{z_{1-\alpha/2}+z_{1-\gamma}}\right)^2 n_2 - 4\tau^2} \tag{39.14}$$

Figure 39.1 shows the power to detect a small relative treatment effect in a two-sided test with a type I error rate of $\alpha = 0.05$ as a function of the cluster size n_1, number of clusters n_2, and the intra-class correlation coefficient ρ. As is obvious, more clusters, larger cluster sizes and a lower intra-class correlation lead to higher power levels. For instance, 114 clusters are needed to achieve a power of 0.8 when there are 10 persons per cluster and the intra-class correlation coefficient is equal to $\rho = 0.05$. For a cluster size of $n_1 = 30$ only 66 clusters are needed. However, the total sample size for the first scenario ($n_1 n_2 = 1140$) is smaller than that for the second ($n_1 n_2 = 1980$). So, the first scenario is favorable when the aim is to minimize the total sample size, whereas the second should be selected when the aim is to minimize the number of clusters, provided that enough clusters with 30 persons are available.

As follows from the left pane in Fig. 39.1 the power increases to one when the number of clusters increases and the cluster size is fixed. On the other hand the power increases to a value not necessarily equal to one when the cluster size increases, given a fixed number of clusters. This can be explained by the fact that the cluster size n_1 appears in both the numerator and denominator of the $\text{var}(\hat{\beta}_1)$, which is inversely related to power, whereas the number of clusters n_2 appears in both. So

$$\lim_{n_1\to\infty} \text{var}(\hat{\beta}_1) = \lim_{n_1\to\infty} 4\frac{\sigma^2+n_1\tau^2}{n_1 n_2} = 4\frac{\tau^2}{n_2} , \tag{39.15}$$

and

$$\lim_{n_2\to\infty} \text{var}(\hat{\beta}_1) = \lim_{n_2\to\infty} 4\frac{\sigma^2+n_1\tau^2}{n_1 n_2} = 0 , \tag{39.16}$$

which explains why a low number of clusters cannot be compensated by a larger cluster size in order to achieve sufficient power.

When both n_1 and n_2 are free to vary, the optimal sample sizes are calculated such that the costs C for

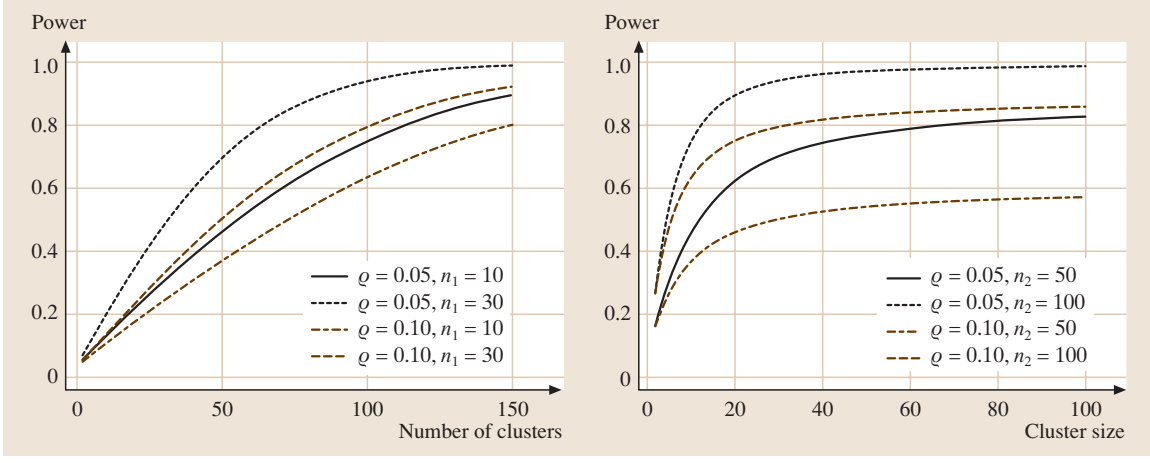


Fig. 39.1 Power as a function of cluster size, number of clusters, and intra-class correlation

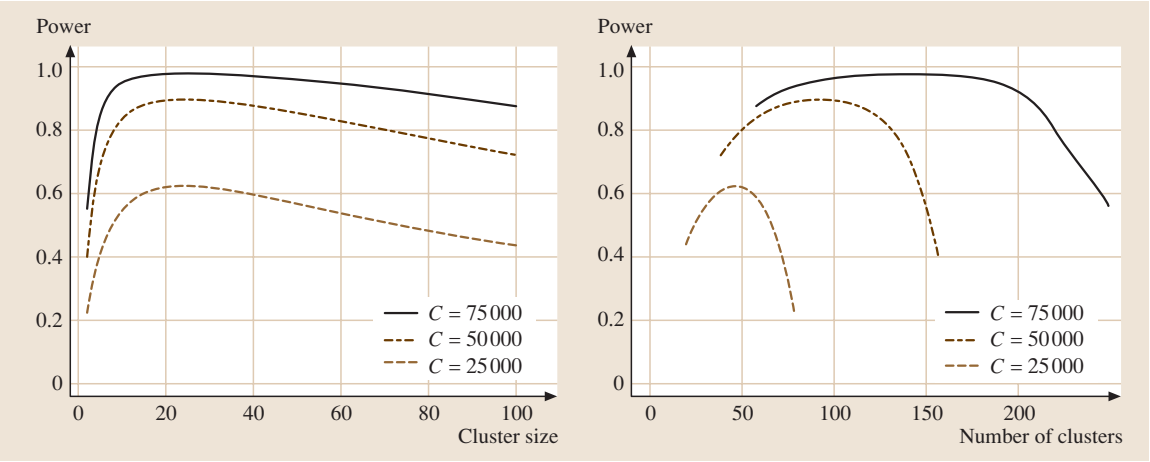


Fig. 39.2 Power as a function of cluster size and number of clusters for various budgets C and costs $c_1 = 300$ and $c_2 = 10$

recruiting and measuring persons and clusters are minimized. These costs are a function of the total number of persons $n_1 n_2$, the number of clusters n_2, the costs per person c_1, and the costs per cluster c_2:

$$C = c_1 n_1 n_2 + c_2 n_2 \,. \tag{39.17}$$

Since the design is balanced, c_1 and c_2 are the costs at the person and cluster level averaged over the two treatment conditions. In general the costs at the cluster level will be much higher than the costs at the person level. The optimal cluster size can be shown to be equal to

$$n_1 = \sqrt{\frac{c_2(1-\rho)}{c_1 \rho}} \,. \tag{39.18}$$

and n_2 follows from (39.13). Equation (39.18) was derived by expressing n_2 in terms of n_1 and C using (39.17), substituting in (39.5) for $\text{var}(\hat{\beta}_1)$, and minimizing with respect to n_1. In some cases the optimal number of persons per cluster is larger than the actual number of persons per cluster. Then, all persons have to be sampled, and additional money should be spent on sampling more clusters.

39.3.2 Maximizing Power Given a Fixed Budget

Equation (39.18) gives the optimal cluster size to achieve a pre-specified power level while minimizing costs C. On the other hand, we can also calculate the optimal cluster size for maximizing the power level when the budget is fixed to a constant C. The optimal cluster size is again equal to that given in (39.18), and the optimal number of clusters is equal to

$$n_2 = \frac{C}{\sqrt{\frac{1-\rho}{\rho} c_1 c_2} + c_2} \,. \tag{39.19}$$

The variance of the treatment effect estimator can be calculated by substituting the optimal n_1 and n_2 from (39.18) and (39.19) into (39.5), which gives

$$\text{var}(\hat{\beta}_1) = (\sigma^2 + \tau^2) \frac{\left[\sqrt{\rho c_2} + \sqrt{(1-\rho) c_1}\right]^2}{C} \,. \tag{39.20}$$

As is obvious, a larger budget C results in a smaller optimal $\text{var}(\hat{\beta}_1)$. Furthermore, a larger budget C results in sampling more clusters, but not in sampling more persons per cluster since the optimal cluster size does not depend on C. The optimal cluster size is an increasing function of the intra-class correlation coefficient ρ, so that larger cluster sizes are required when there is much variation in the outcome at the person level. Furthermore, the optimal cluster size is a function of the costs c_2 for recruiting a cluster relative to the costs c_1 for sampling a person. So, fewer clusters will be sampled in favor of sampling more persons per cluster when it is relatively expensive to sample a cluster.

Figure 39.2 shows the power to detect a small treatment effect as a function of the cluster size, number of clusters and total budget C when $c_1 = 300$ and $c_2 = 10$ and $\rho = 0.05$. The optimal cluster size is $n_1 = 24$ and this value does not depend on the budget. A budget approximately equal to $C = 75\,000$ is required to achieve a power level of 0.9 to detect a small treatment effect. The optimal number of clusters is an increasing function

of the budget C. For a large budget the power curve is rather flat around it optimum, but this is not the case for lower budgets. Of course, these power curves hold when dropout is absent, and a somewhat larger sample size is required when persons and/or clusters are expected to drop out.

39.4 The Effect of Adding Covariates

Until now we have only considered optimal designs for models without covariates. This section focuses on the effects of adding a single covariate x_{ij} that varies at the cluster and/or person level on the optimal sample sizes. The extension to multiple covariates is straightforward and not given here. The between- and within-cluster effect of the covariate on the outcome are not necessarily the same [39.20]. The covariate is therefore split up into a between-cluster component $\bar{x}_{.j}$ and a within-cluster component $(x_{ij}-\bar{x}_{.j})$, and the multilevel model is given by

$$y_{ij}=\beta_0^*+\beta_1^* z_j+\beta_2^*\bar{x}_{.j}+\beta_3^*(x_{ij}-\bar{x}_{.j})+u_j^*+e_{ij}^*\,, \tag{39.21}$$

where $\beta_2^*\neq\beta_3^*$. As in the model without covariates, the random terms $u_j^*\sim N(0,\tau^{*2})$ and $e_{ij}^*\sim N(0,\sigma^{*2})$ are assumed to be independent of each other and the covariate.

When the covariate only varies at the cluster level, the term $\beta_3^*(x_{ij}-\bar{x}_{.j})$ is equal to zero and may be removed from model. An example of a cluster-level covariate is the type of school (public versus private) in a school-based smoking prevention intervention. Likewise, when the covariate only varies at the person level, the term $\beta_2^*\bar{x}_{.j}$ is equal to zero and may be removed from the model. An example of such a covariate is gender, given that the percentage of boys per school does not vary across the schools.

Note that the regression coefficients and random terms are superscripted with an asterisk in order to stress that their values may differ from those in the model without covariates (39.1). Given a grand-mean centered covariate and treatment condition coded $z_j=-0.5$ for the control group and $z_j=+0.5$ for the intervention group, the treatment effect is estimated by

$$\hat{\beta}_1^*=\frac{\sum z_j y_{ij}\sum x_j^2-\sum z_j x_j\sum x_j y_{ij}}{n_1 n_2\sum x_j^2(1-r_{xc}^2)}\,, \tag{39.22}$$

with variance

$$\mathrm{var}(\hat{\beta}_1^*)=4\frac{\sigma^{*2}+n_1\tau^{*2}}{n_1 n_2}\frac{1}{(1-r_{zx}^2)}\,. \tag{39.23}$$

When comparing formulae (39.23) with that for the variance in a model without covariates, we see that an additional factor $1/(1-r_{zx}^2)$ is introduced. This factor is often called the variance inflation factor (VIF), and $\mathrm{var}(\hat{\beta}_1^*)$ reaches it minimum when the correlation r_{zx}^2 between the treatment condition and covariate is equal to zero. The within-cluster component $(x_{ij}-\bar{x}_{.j})$ and the treatment condition z_j are orthogonal, and therefore r_{zx}^2 is equal to the correlation between the between-cluster component $\bar{x}_{.j}$ and the treatment condition z_j. For normally and binary covariates this correlation is approximately normally distributed with variance $1/n_2$ [39.21], and thus $r_{zx}^2\in(0,4/n_2)$ with 95% probability. So, this correlation will be small when the number of clusters is large, and clusters are randomly assigned to treatment conditions. When the cluster randomized trial only has a small number of clusters, a correlation r_{zx}^2 equal to zero may be achieved by pre-stratification on the covariate, which means that for each value of $\bar{x}_{.j}$ half of the clusters are randomized to the control condition while the others are randomized to the intervention condition.

In the remainder of this section we will assume that the correlation between covariate and treatment condition is zero. Then, the estimated treatment effect is equal to that in a model without covariates, and the optimal sample sizes are equal to those in a model without covariates as given in (39.18) and (39.19) with τ^2 and σ^2 replaced by τ^{*2} and σ^{*2}, respectively [39.22]. The relations between the variance components in a model with and without covariates can be established using the method described in [39.23]. During the analysis stage the total variation in the outcome y_{ij} is given by the observed data and the estimated variance components change if covariates are added to or excluded from the model. The chance in the estimated variance components can be derived by assuming that the variance of the observed outcomes and covariance of two outcomes within the same cluster are given by the data and are therefore equal for model (39.21) and (39.1):

$$\begin{aligned}\mathrm{var}(y_{ij})&=\mathrm{var}(\beta_1 z_j+u_j+e_{ij})\\&=\mathrm{var}\left[\beta_1^* z_j+\beta_2^*(x_{ij}-\bar{x}_{.j})+\beta_3^*\bar{x}_{.j}+u_j^*+e_{ij}^*\right]\end{aligned} \tag{39.24}$$

Table 39.2 Changes in the variance components due to the inclusion of a covariate

Changes due to the inclusion of $\bar{x}_{.j}$	Changes due to the inclusion of $(x_{ij} - \bar{x}_{.j})$
$\tau^2 - \tau^{*2} = \hat{\beta}_2^{*2}\text{var}(\bar{x}_{.j}) > 0$	$\tau^2 - \tau^{*2} = \hat{\beta}_3^{*2}\text{cov}(x_{ij} - \bar{x}_{.j}, x_{i,j} - \bar{x}_{.j}) < 0$ ≈ 0 for large n_1
$\sigma^2 - \sigma^{*2} = 0$	$\sigma^2 - \sigma^{*2} = \hat{\beta}_3^{*2}\left[\text{var}(x_{ij} - \bar{x}_{.j}) - \text{cov}(x_{ij} - \bar{x}_{.j}, x_{i,j} - \bar{x}_{.j})\right] > 0$ $\approx \hat{\beta}_3^{*2}\text{var}(x_{ij} - \bar{x}_{.j}) > 0$ for large n_1
Note: It is assumed that $r_{zx}^2 = 0$	

and

$$\begin{aligned}\text{cov}(y_{ij}, y_{i'j}) &= \text{cov}(\beta_1 z_j + u_j, \beta_1 z_j + u_j)\\ &= \text{cov}\Big[\beta_1^* z_j + \beta_2^*(x_{ij} - \bar{x}_{.j}) + \beta_3^* \bar{x}_{.j}\\ &\quad + u_j^*, \beta_1^* z_j + \beta_2^*(x_{i'j} - \bar{x}_{.j})\\ &\quad + \beta_3^* \bar{x}_{.j} + u_j^*\Big]\,. \end{aligned} \tag{39.25}$$

Table 39.2 shows the changes in the estimated variance components due to the inclusion of one covariate. The variance component at the person level remains unchanged when a cluster-level covariate is added to the model, and decreases when a person-level covariate is added to the model. The variance component at the cluster level decreases when a cluster-level component is added, but *increases* when a person-level covariate is added. However, for large cluster sizes this increase is negligible, and it may be nullified by the decreasing effect of adding a cluster-level covariate. So, adding covariates will in general lead to a decrease in the variance components, and therefore in a more efficient design, given a zero correlation between the covariate and treatment condition.

Of course, costs are associated with measuring covariates and one may wonder when adding a covariate may be a more cost-efficient strategy to increase the power to detect a treatment effect than sampling more clusters. Both strategies have recently been compared, and it was concluded that adding covariates is more efficient when the costs to measure these covariates are small and the correlation between the covariate and the outcome is large [39.24]. Adding a covariate at the cluster level is recommended when clusters are large (say $n_1 = 100$) and the costs to recruit and measure a cluster are small in relation to the costs to recruit and measure a person. Vice versa, adding a covariate that only varies at the person level is recommended when clusters are small (say $n_1 = 4$) and the relative costs to recruit and measure a cluster are large.

39.5 Robustness Issues

In the Sect. 39.3 it was shown that the optimal sample sizes depend on the value of the intra-class correlation coefficient. The value of this parameter is generally unknown at the design stage and an educated guess must be obtained from subject-matter knowledge or similar studies in the past. Table 1 in [39.25] gives an overview of recent papers that report values of the intra-class correlation coefficient. There is, however, no guarantee that the values of similar studies in the past are the true values for the current study at hand, since the study may be conducted in a different year of country, or may target a different population (e.g. elementary-school children instead of high-school children).

As an example consider a cluster randomized trials that aims at detecting a small relative treatment effect at power level 0.9 in a two-sided test with $\alpha = 0.05$. The cluster size is equal to $n_1 = 30$, and the true but unknown intra-class correlation is $\rho = 0.05$. The required number of clusters at prior value $\rho = 0.05$ is equal to $n_2 = 86$, and this results in a power equal to 0.9, since the prior ρ is equal to the true ρ. However, if the prior estimate is equal to $\rho = 0.10$, then the required number of clusters can be calculated to be equal to $n_2 = 138$. Thus, the number of clusters is overestimated by 60%, and the power level at the true ρ is equal to 0.98. For a prior estimate as small as $\rho = 0.025$, the required number of clusters is equal to $n_2 = 62$, which results in a power level of 0.78 at the true ρ. Hence, cluster randomized trials are not very robust against an incorrect prior estimate of the intra-class correlation coefficient.

Since it is increasingly difficult to obtain adequate financial recourses, and since cluster randomized trials require the willingness of clusters and persons to participate, it is extremely important to design trials such

Table 39.3 Assumptions about the intra-class correlation coefficient, with associated power with 86 groups and required number of groups for a power level of 0.9

Intra-class correlation coefficient Median (95% interval)	Power with 86 groups Median (95% interval)	Number of groups for power = 0.9 Median (95% interval)
0.05–0.051	0.90–0.898	86–88
0.008–0.099	0.734–0.995	44–136

that they are not under- or overpowered. Two procedures to calculate robust optimal designs are Bayesian optimal designs, where a prior distribution on the intra-class correlation is used, and designs with sample-size re-estimation based on data obtained from a pilot.

39.5.1 Bayesian Optimal Designs

Bayesian methods allow us to implicitly take uncertainty about model parameters into account by using a prior distribution on the parameters. Consider the example given above and suppose that we assume the intra-class correlation to be around 0.05, but that there is some change that it is up to 0.10. This uncertainty may be reflected by a normal distribution with mean 0.05 and standard deviation 0.025, but truncated at zero so that we exclude negative values. We can now sample from this prior distribution and calculate the required number of clusters to achieve a power level of 0.9. In addition, we can also calculate the power level that is achieved with 86 clusters.

The results in Fig. 39.3 and Table 39.3 were obtained after 100 000 iterations, which took less that one minute on a desktop computer with a 2.8-GHz CPU and 1 Gb of RAM. The median intra-class correlation coefficient is equal to 0.051, at which there are hardly any values larger than 0.1. The median power achieved with 86 clusters is equal to 0.0898, so there is a change of about 50% that the power is less than the required level of 0.9. In some cases, it can even be as small as 0.7. The median required number of clusters is equal to 88, whereas the boundaries of the 95% interval are 44 and 136. So, on the basis of the results in Fig. 39.3 and Table 39.3 we might decide to use a number of clusters larger than 86 to be reasonably confident that the study has sufficient power.

39.5.2 Designs with Sample-Size Re-Estimation

Designs with sample-size re-estimation have been proposed by *Stein* [39.26] in the context of comparing two treatment conditions with respect to a continuous outcome. His procedure includes two stages. In the first stage (the internal pilot) the variance of the outcome is estimated using the observations collected so far, and the total sample size is re-estimated based on the variance estimate. In the second stage the remainder of the observations is collected such that re-estimated total sample size is achieved. Only the observations of the first stage are used to estimate the variance

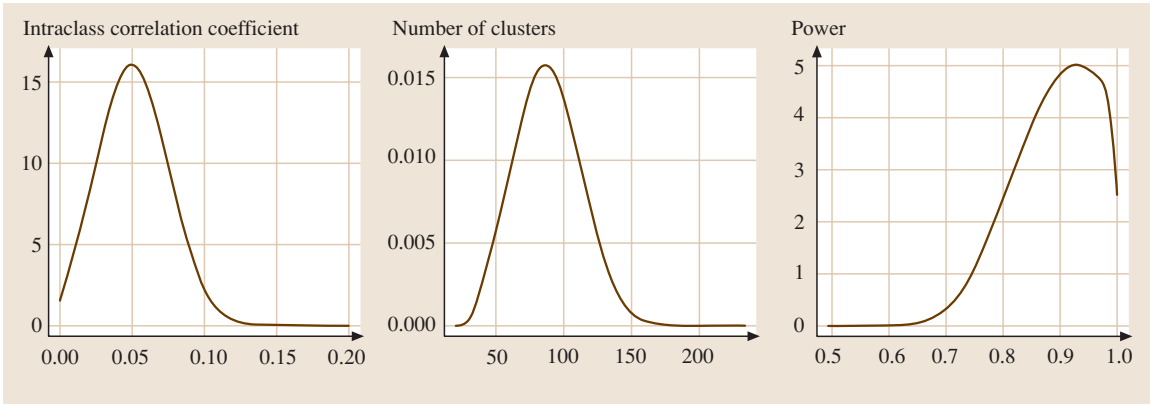


Fig. 39.3 Densities of the prior distribution of the intra-class correlation coefficient, the required number of clusters to achieve a power level 0.9, and the power at 86 clusters

Table 39.4 Empirical type I error rate α and power $1-\beta$ for the standard design and re-estimation design for three values of the prior ρ. The true $\rho = 0.05$

Prior ρ	Standard design		Re-estimation design					
			$\pi = 0.25$		$\pi = 0.5$		$\pi = 0.75$	
	α	$1-\beta$	α	$1-\beta$	α	$1-\beta$	α	$1-\beta$
0.025	0.0538	0.7812	0.0526	0.8690	0.0600	0.9072	0.0586	0.9012
0.05	0.0480	0.9004	0.0576	0.8886	0.0530	0.9094	0.0556	0.8986
0.10	0.0502	0.9832	0.0534	0.8964	0.0588	0.9114	0.0532	0.9474

of the outcomes, while all observations are used in the calculation of the group means. This procedure was modified by *Wittes* and *Britain* [39.27] such that all data are used in the final analysis. In contrast to the Stein procedure, the Wittes and Britain procedure does not preserve the type I error rate since the total sample size depends on the variance estimate in the pilot. Internal pilots have been shown to work well for cluster randomized trials by *Lake* et al. [39.28].

We consider the same example where we wish to detect a small relative treatment effect at power level 0.9. The true $\rho = 0.05$, and we have three prior values $\rho = 0.025$, 0.05, and 0.10. Table 39.4 shows the empirical type I error rates and power levels in a simulation study with 5000 runs. The power levels for the design without sample-size re-estimation (i. e. the standard design) are too small when the prior ρ is underestimated and too large when the prior ρ is overestimated. The values of the type I error rate are close to their nominal value of $\alpha = 0.05$.

For designs with sample-size re-estimation the required number of clusters is calculated on the basis of the prior ρ . Then, a predefined proportion π of this number of clusters is used in the internal pilot. The required number of clusters in the second stage is calculated on the basis of the parameter estimates obtained from data collected in the internal pilot. When the size of the internal pilot is already sufficiently large, a second stage is not needed. Table 39.4 shows that the power levels for studies with incorrect prior values ρ are much closer to the value 0.9 than they are in the standard design. For $\pi = 0.25$ and prior $\rho = 0.025$, the power is somewhat lower than 0.9, which is explained by the fact that the size of the internal pilot is somewhat too small to result in a good estimate of the true ρ. For $\pi = 0.75$ and prior $\rho = 0.10$, the power is larger than 0.9, which is explained by the fact that the size of the internal pilot is already too large. The empirical type I error rates are somewhat, but not dramatically, larger than the nominal value $\alpha = 0.05$.

39.6 Optimal Designs for the Intra-Class Correlation Coefficient

So far we have focussed on optimal designs that maximize the power to detect a treatment effect or, equivalently, minimize the variance of the treatment effect estimator. Another option is to design a study such that it minimizes the variance of the intra-class correlation coefficient estimator, which is equal to

$$\mathrm{var}(\hat{\rho}) = \frac{2(1-\rho)^2(1+(n_1-1)\rho)^2}{(n_1-1)(n_1n_2-n_1)} \,. \qquad (39.26)$$

Such optimal designs are especially useful for pilot studies that aim at an estimate of the intra-class correlation coefficient. Again, we can minimize this variance under the precondition that the costs for recruiting persons and clusters do not exceed the budget, as specified by (39.17). Closed-form equations for the optimal n_1 and n_2 do not exist. Instead, the optimal design may be found by expressing n_2 in terms of n_1, c_1, c_2 and C using (39.17): $n_2 = C/(c_1n_1 + c_2)$. This relation may then be substituted into (39.26), from which the optimal n_1 may be calculated.

For most trials the main focus lies on the treatment effect, but researchers may also be interested in the degree of variability of the outcome that is between clusters. If the amount of between-cluster variability turns out to be high, then one may wish to identify those schools for which the intervention performs worst and try to characterize these schools in terms of their school-level variables. The intervention can then be adjusted for these types of schools. For instance, a smoking prevention intervention that works well for high schools may

have to be adjusted for schools for lower vocational education.

When a researcher has multiple objectives in mind, he or she may design a multiple-objective optimal design. Suppose that we wish to design a trial that aims at estimating both the treatment effect and intra-class correlation with largest precision, that is, it aims at minimizing $\text{var}(\hat{\beta}_1)$ and $\text{var}(\hat{\rho}_1)$. These two variances are the two objectives and the first is the most important since the trial is, in the first instance, designed to gain insight into the value of the treatment effect, whereas the intra-class correlation coefficient is of secondary importance. The two-objective optimal design is the one that does best under the criterion $\text{var}(\hat{\rho}_1)$ subject to the constraint that the value $\text{var}(\hat{\beta}_1)$ is smaller than a user-specified constant c:

$$\min \text{var}(\hat{\rho}) \text{ subject to } \text{var}(\hat{\beta}_1) \leq c \,. \tag{39.27}$$

The design that satisfies this criterion is often called a constrained optimal design. For convenience, this criterion is often rewritten as

$$\min \text{var}(\hat{\rho}) \text{ subject to } \text{eff}(\hat{\beta}_1) \geq e \,, \tag{39.28}$$

where $\text{eff}(\hat{\beta}_1)$ is the efficiency in estimating the treatment effect. So, the least important optimality criterion is minimized subject to the constraint that the efficiency in estimating the treatment effect is larger than a user-selected constraint. The efficiency is calculated as the $\text{var}(\hat{\beta}_1)$ obtained with the optimal sample sizes as given by (39.18) and (39.19) divided by the $\text{var}(\hat{\beta}_1)$ obtained with any other sample sizes n_1 and n_2. The efficiency varies between zero and one. Its interpretation is that, if N observations are used in the optimal design, then $N/\text{eff}(\hat{\beta}_1)$ observations are used in the suboptimal design to obtain the same amount of information.

Constrained optimal designs are often difficult to derive, and one may wish to construct a compound optimal design to minimize

$$\lambda \, \text{var}(\hat{\rho}) + (1-\lambda) \, \text{var}(\hat{\beta}_1) \,. \tag{39.29}$$

Compound optimal designs are generally easier to solve, either numerically or analytically. Under convexity and differentiability constrained and compound optimal designs are equivalent [39.29]. So, in order to derive

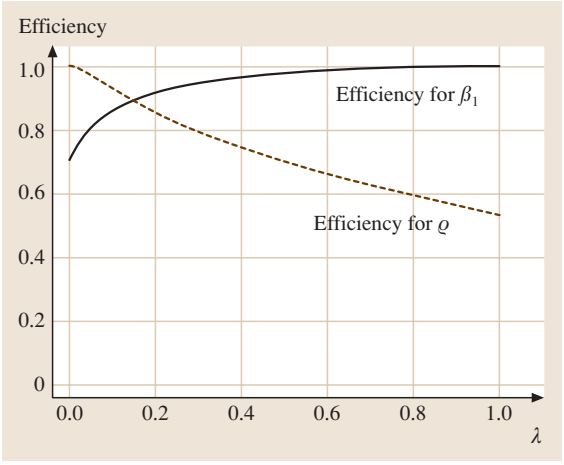


Fig. 39.4 Efficiency plot

the constrained optimal design one may first derive the compound optimal design as a function of the weight λ in (39.29). That is, for each value of λ the sample sizes n_1 and n_2 that minimize (39.29) are derived. Subsequently, an efficiency plot is drawn in which the efficiencies $\text{eff}(\hat{\beta}_1)$ and $\text{eff}(\hat{\rho})$ are plotted as a function of λ. The constrained optimal design is the design for which $\text{eff}(\hat{\beta}_1) \geq e$ and $\text{eff}(\hat{\rho})$ is maximized. In most practical situations the constant e is chosen to be 0.8 or 0.9.

Figure 39.4 shows an efficiency plot for a trial with $C = 50\,000$, $c_1 = 30$, $c_2 = 10$ and $\rho = 0.025$. The optimal design for estimating ρ with the greatest precision is $n_1 = 45.4$ and $n_2 = 103.4$ and is achieved when $\lambda = 1$. The efficiency for ρ is a decreasing function of λ. The optimal for estimating β_1 with largest precision is $n_1 = 10.8$ and $n_2 = 362.8$, and is achieved when $\lambda = 0$. The efficiency for β_1 is a increasing function of λ. Note that the two lines do not necessarily meet at the point. When we wish to estimate ρ with the greatest precision, given the condition that $\text{eff}(\hat{\beta}_1) \geq 0.9$, then we draw a horizontal line at $e = 0.9$ to intersect the graph of $\text{eff}(\hat{\beta}_1)$. Then a vertical line is drawn from this point of intersection to meet the λ-axis. This results in $\lambda = 0.17$, which corresponds to $n_1 = 23.42$ and $n_2 = 189.3$, and $\text{eff}(\hat{\rho}) = 0.876$. Of course, these sample sizes have to rounded off to integer values. Large efficiencies are possible for both criteria, which are therefore called compatible.

39.7 Conclusions and Discussion

Cluster randomized trials randomize complete groups of persons, rather than the persons themselves, to treatment conditions. They are often used in situations where the intervention is delivered to groups of persons, such as in school-based smoking prevention interventions with class teaching on smoking and health. Since the outcomes of persons in a group cannot be considered to be independent, a larger sample size is required to achieve a pre-specified power level than in a simple randomized trial, especially when the intra-class correlation coefficient and/or the cluster size are large.

Multisite trials are an alternative to cluster randomized trials. Multisite trials randomize persons within clusters to treatment conditions, such that all treatments are available within each cluster. So, for multisite trials cluster and treatment condition are crossed, whereas for cluster randomized trails clusters are nested within treatment conditions. Multisite trials have two advantages above cluster randomized trials: they are more powerful, and they allow for the estimation of the cluster by treatment interaction [39.30]. A main drawback of multisite trials is that they do not protect from control-group contamination, which occurs when information on the intervention leaks from the individuals in the intervention group to those in the control group [39.31]. In some cases blinding may be an option to prevent control-group contamination, such as in double-blind placebo-controlled multicentre clinical trial with patients nested within clinics. This is an option when the experimental treatment is a new pill, which only differs from the pills in the control group by the amount of active substance. When patients are randomly assigned to treatment conditions and neither the patient nor the researchers know who belongs to which treatment, a multisite study may be an alternative to a cluster randomized trial. Blinding is of course no option when the intervention consists of interpersonal relationships, such as in peer-pressure groups. Control-group contamination may also be due to the person delivering the intervention, such as in guideline trials with patients nested within family practices. If both a control and intervention group are available in each practice, it will be extremely difficult for the physician not to let patients in the control group benefit from the intervention. Of course, the choice for a cluster randomized trial does not guarantee the absence of control-group contamination. An example is a trial in which general practices are randomized to treatment conditions and the intervention consists of leaflets to promote healthy lifestyles. Control-group contamination can occur when staff members work between practices and distribute leaflets in the control practices. Another example is a school-based smoking prevention intervention where children from different families attend different schools, and thereby encounter different treatment conditions.

This chapter has given an introduction to the design and analysis of cluster randomized trails. It focused on models with two levels of nesting, two treatment conditions, and continuous outcomes. The extension to three or more levels of nesting is straightforward and can be found elsewhere [39.30, 32]. The optimal sample sizes were shown to depend on the value of the intra-class correlation coefficient and it was shown that an incorrect prior may lead to an under- or overpowered study. This may be overcome by using a robust optimal design, such as a Bayesian optimal design or a design using sample-size re-estimation. Such designs are also very useful for the planning of cluster randomized trials with binary outcomes, since then the optimal sample size can be shown not only to depend on the intra-class correlation coefficient, but also on the probabilities of a positive response in each treatment condition [39.33, 34].

References

39.1 A. Sommer, I. Tarwotjo, E. Djunaedi, K. P. West, A. A. Loeden, R. Tilden, L. Mele: Impact of vitamin A supplementation on childhood mortality. A randomised controlled community trial, Lancet **1986**, 1169–1173 (1986)

39.2 M. Moerbeek, G. J. P. van Breukelen, M. P. F. Berger: A comparison between traditional methods and multilevel regression for the analysis of multicenter intervention studies, J. Clin. Epidemiol. **56**, 341–350 (2003)

39.3 H. Goldstein: *Multilevel Statistical Models*, 3rd edn. (Edward Arnold, London 2003) p. 3

39.4 J. Hox: *Multilevel Analysis. Techniques and Applications* (Erlbaum, New Jersey 2002)

39.5 T. A. B. Snijders, R. J. Bosker: *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling* (Sage, London 1999)

39.6 N. T. Longford: *Random Coefficient Models* (Clarendon, Oxford 1993)

39.7 S. W. Raudenbush, A. S. Bryk: *Hierarchical Linear Models. Applications and Data Analysis Methods* (Sage, Thousand Oaks 2002)

39.8 S. R. Searle, G. Casella, C. E. McCulloch: *Variance Components* (Wiley, New York 1992)

39.9 S. W. Raudenbush: Hierarchical linear models and experimental design. In: *Applied Analysis of Variance in Behavioral Science*, ed. by L. K. Edwards (Wiley, New York 1993) pp. 459–496

39.10 H. Goldstein: Multilevel mixed linear model analysis using iterative generalized least squares, Biometrika **73**(1), 43–56 (1986)

39.11 H. Goldstein: Restricted unbiased iterative generalized least squares estimation, Biometrika **76**, 622–623 (1989)

39.12 N. T. Longford: A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects, Biometrika **74**, 817–827 (1987)

39.13 A. P. Dempster, D. B. Rubin, R. K. Tsutakawa: Estimation in covariance components models, J. Am. Stat. Assoc. **76**(374), 341–353 (1981)

39.14 W. M. Mason, G. Y. Wong, B. Entwisle: Contextual analysis through the multilevel linear model. In: *Sociological Methodology 1983–1984*, ed. by S. Leinhardt (Jossey-Bass, San Francisco 1983) pp. 72–103

39.15 J. Rasbash, F. Steele, W. Browne: *A User's Guide to MLwiN Version 2.0* (Institute of Education, London 2004)

39.16 S. W. Raudenbush: *HLM 6. Hierarchical Linear and Nonlinear Modeling* (Scientific Software International, Chicago 2004)

39.17 J. Cohen: A power primer, Psychol. Bull. **112**(1), 155–159 (1992)

39.18 W. G. Cochran: *Planning and Analysis of Observational Studies* (Wiley, New York 1983)

39.19 A. K. Manatunga, M. G. Hudges, S. Chen: Sample size estimation in cluster randomized studies with varying cluster size, Biom. J **43**(1), 75–86 (2001)

39.20 J. M. Neuhaus, J. D. Kalbfleisch: Between- and within-cluster covariate effects in the analysis of clustered data, Biometrics **54**, 638–645 (1998)

39.21 M. Kendall, A. Stuart: *The Advanced Theory of Statistics. Vol. 2: Inference and Relationship* (Griffin, London 1979)

39.22 M. Moerbeek, G. J. P. van Breukelen, M. P. F. Berger: Optimal experimental designs for multilevel models with covariates, Commun. Statist. Theory Methods **30**(12), 2683–2697 (2001)

39.23 T. A. B. Snijders, R. J. Bosker: Modeled variance in two-level models, Sociol. Methods Res. **22**(3), 342–363 (1994)

39.24 M. Moerbeek: Power and money in cluster randomized trials: when is it worth measuring a covariate?, Stat. Med. in press

39.25 D. M. Murray, S. P. Varnell, J. L. Blitstein: Design and analysis of group-randomized trials: A review of recent methodological developments, Am. J. Public Health **94**(3), 423–432 (2004)

39.26 A. C. Stein: A two-sample test for a linear hypothesis whose power is independent of the variance, Ann. Math. Stat. **29**, 1271–1275 (1945)

39.27 J. Wittes, E. Brittain: The role of internal pilot studies in increasing the efficiency of clinical trials, Stat. Med. **9**(1), 65–72 (1990)

39.28 S. Lake et al.: Sample size re-estimation in cluster randomization trials, Stat. Med. **21**(10), 1337–1350 (2002)

39.29 D. Cook, W. K. Wong: On the equivalence of constrained and compound optimal designs, J. Am. Stat. Assoc. **89**(426), 687–692 (1994)

39.30 M. Moerbeek, G. J. P. van Breukelen, M. P. F. Berger: Design issues for experiments in multilevel populations, J. Educ. Behav. Stat. **25**(3), 271–284 (2000)

39.31 M. Moerbeek: Randomization of clusters versus randomization of persons within clusters: Which is preferable?, Am. Stat. **59**(1), 72–78 (2005)

39.32 T. C. Headrick, B. D. Zumbo: On optimizing multilevel designs: Power under budget constraints, Austr. New Zealand J. Stat. **47**(2), 219–229 (2005)

39.33 M. Moerbeek, G. J. P. van Breukelen, M. P. F. Berger: Optimal experimental design for multilevel logistic models, Statistician **50**(1), 17–30 (2001)

39.34 M. Moerbeek, C. J. M. Maas: Optimal experimental designs for multilevel logistic models with two binary predictors, Commun. Stat. Theory Methods **34**(5), 1151–1167 (2005)

40. A Two-Way Semilinear Model for Normalization and Analysis of Microarray Data

A proper normalization procedure ensures that the normalized intensity ratios provide meaningful measures of relative expression levels. We describe a two-way semilinear model (TW-SLM) for normalization and analysis of microarray data. This method does not make the usual assumptions underlying some of the existing methods. The TW-SLM also naturally incorporates uncertainty due to normalization into significance analysis of microarrays. We propose a semiparametric M-estimation method in the TW-SLM to estimate the normalization curves and the normalized expression values, and discuss several useful extensions of the TW-SLM. We describe a back-fitting algorithm for computation in the model. We illustrate the application of the TW-SLM by applying it to a microarray data set. We evaluate the performance of TW-SLM using simulation studies and consider theoretical results concerning the asymptotic distribution and rate of convergence of the least-squares estimators in the TW-SLM.

Microarrays are a useful tool for monitoring gene expression levels on a large scale and has been widely used in functional genomics [40.1, 2]. In a microarray experiment, cDNA segments representing the collection of genes and expression sequence tags (ESTs) to be probed are amplified by the polymerase chain reaction (PCR) and spotted in high density on glass microscope slides using a robotic system. Such slides are called microarrays. Each microarray contains thousands of reporters of the collection of genes or ESTs. The microarrays are queried in a co-hybridization assay using two fluorescently labeled biosamples prepared from the cell populations of interest. One sample is labeled with the fluorescent dye Cy5 (red), and another with the fluorescent dye Cy3 (green). Hybridization is assayed using a confocal laser scanner to measure fluorescence intensities, allowing simultaneous determination of the relative expression levels of all the genes represented on the slide [40.3]. The ability to monitor gene expressions on a large scale promises to have a profound impact on the understanding of basic cellular processes, developing better tools for disease diagnostics and treatment, cancer classification, and identifying drug targets, among others. Indeed, microarrays have already been used for detecting differentially expressed genes in different cell populations, classifying different cancer subtypes, identifying gene clusters based on co-expressions [40.4–7].

Because a microarray experiment monitors thousands of genes simultaneously, it routinely produces a massive amount of data. This and the unique nature of microarray experiments present a host of challenging statistical issues. Some of these can be dealt with using the existing statistical methods, but many are novel questions that require innovative solutions. One such question is normalization. The purpose of normalization is to remove bias in the observed expression levels and establish the baseline ratios of intensity levels from

the florescent dyes Cy3 and Cy5 across the whole dynamic range. A proper normalization procedure ensures that the intensity ratios provide meaningful measures of relative expression levels. In a microarray experiment, many factors may cause bias in the observed expression levels, such as differential efficiency of dye incorporation, differences in concentration of DNA on arrays, difference in the amount of RNA labeled between the two channels, uneven hybridizations, differences in the printing pin heads, among others.

Many researchers have considered various normalization methods; see for example [40.8–13]. For reviews of some of the existing normalization methods, see [40.14, 15]. More recently, *Fan* et al. [40.16] proposed a semilinear in-slide model (SLIM) method that makes use of replications of a subset of the genes in an array. If the number of replicated genes is small, the expression values of the replicated genes may not cover the entire dynamic range or reflect the spatial variation in an array. *Fan* et al. [40.17] generalized the SLIM method to account for across-array information, resulting in an aggregated SLIM, so that replication within an array is no longer required.

A widely used normalization method is the local regression *lowess* [40.18] normalization proposed by *Yang* et al. [40.11]. This method estimates the normalization curves using the robust *lowess* for log-intensity ratio versus log-intensity product using all the genes in the study. The underlying assumption of this normalization method is either that the number of differentially expressed genes is relatively small or that the expression levels of up- and down-regulated genes are symmetric, so that the *lowess* normalization curves are not affected significantly by the differentially expressed genes. If it is expected that many genes will have differential expressions, *Yang* et al. [40.11] suggested using dye-swap for normalization. This approach makes the assumption that the normalization curves in the two dye-swaped slides are symmetric. Because of the slide-to-slide variation, this assumption may not always be satisfied.

Strictly speaking, an unbiased normalization curve should be estimated using genes whose expression levels remain constant and cover the whole range of the intensity. Thus *Tseng* et al. [40.12] first used a rank-based procedure to select a set of invariant genes that are likely to be non-differentially expressed, and then use these genes for *lowess* normalization. However, they pointed out that the number of invariant genes may be small and not cover the whole dynamic range of the expression values, and extrapolation is needed to fill in the gaps that are not covered by the invariant genes. In addition, a threshold value is required in this rank-based procedure. The level of the sensitivy of the final result to the threshold value may need to be evaluated on a case-by-case basis.

A common practice in microarray data analysis is to consider normalization and detection of differentially expressed genes separately. That is, the normalized values of the observed expression levels are treated as data in the subsequent analysis. However, because normalization typically includes a series of statistical adjustments to the data, there are variations associated with this process. These variations will be inherited in any subsequent analysis. It is desirable to take them into account in order to assess the uncertainty of the analysis results correctly.

We have proposed a two-way semilinear model (TW-SLM) for normalization and analysis of microarray data [40.19–21]. When this model is used for normalization, it does not require some of the assumptions that are needed in the *lowess* normalization method. Below, we first give a description of this model, and then suggest an M-estimation (including the least squares estimator as a special case) and a local regression method for estimation in this model. We describe a back-fitting algorithm for computation in the model. We then consider several useful extensions of this model. We illustrate the application of the TW-SLM by applying it to the Apo A1 data set [40.7]. We evaluate the performance of TW-SLM using simulation studies. We also state theoretical results concerning the asymptotic distribution and rate of convergence of the least squares estimator of the TW-SLM.

40.1 The Two-Way Semilinear Model

Suppose there are J genes and n slides in the study. Let R_{ij} and G_{ij} be the red (Cy 5) and green (Cy 3) intensities of gene j in slide i, respectively. Let y_{ij} be the log-intensity ratio of the red over green channels of the j-th gene in the i-th slide, and let x_{ij} be the corresponding average of the log-intensities of the red and green channels. That is,

$$y_{ij} = \log_2 \frac{R_{ij}}{G_{ij}}, \quad x_{ij} = \frac{1}{2}\log_2(R_{ij}G_{ij}),$$
$$i = 1, \ldots, n, \quad j = 1, \ldots, J\,.$$

Let $z_i \in \mathbb{R}^d$ be a covariate vector associated with the i-th slide. It can be used to code various types of designs. The TW-SLM model decomposes the observed intensity ratio y_{ij} in the following way:

$$y_{ij} = f_i(x_{ij}) + z_i' \beta_j + \sigma_{ij}\varepsilon_{ij}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, J, \tag{40.1}$$

where f_i is the intensity-dependent normalization curve for the i-th slide, $\beta_j \in \mathbb{R}^d$ is the effect associated with the j-th gene; σ_{ij} are the residual standard deviation, ε_{ij} have mean 0 and variance 1. We note that f_i can be considered as the log-intensity ratios in the absence of the gene effects. From a semiparametric modeling standpoint, these f_i functions are nonparametric components in the model and are to be estimated. In model (40.1), it is only made explicit that the normalization curves f_i are slide-dependent. It can also be made dependent upon regions of a slide to account for spatial effects. For example, it is straightforward to extend the model with an additional subscript in (y_{ij}, x_{ij}) and f_i and make f_i also depend on the printing-pin blocks within a slide. We describe this and two other extensions of TW-SLM in Sect. 40.4. Below, we denote the collection of the normalization curves by $\boldsymbol{f} = \{f_1, \ldots, f_n\}$ and the matrix of the gene effects by $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)' \in \mathbb{R}^{J\times d}$. Let $\Omega_0^{J\times d}$ be the space of all $J \times d$ matrices $\boldsymbol{\beta}$ satisfying $\sum_{j=1}^J \beta_j = 0$. From the definition of the TW-SLM model (40.1), $\boldsymbol{\beta}$ is identifiable only up to a member in $\Omega_0^{J\times d}$.

We call (40.1) TW-SLM since it contains the two-way analysis of variation (ANOVA) model as a special case with $f_i(x_{ij}) = \alpha_i$ and $z_i = 1$. Our approach naturally leads to the general TW-SLM

$$y_{ij} = f_i(x_{ij}) + z_{ij}' \beta_j + \varepsilon_{ij}, \tag{40.2}$$

which could be used to incorporate additional prior knowledge in the TW-SLM (Sect. 40.3). The identifiability condition $\sum_j \beta_j = 0$ is no longer necessary in (40.2) unless $z_{ij} = z_i$ as in (40.1).

The TW-SLM is an extension of the semiparametric regression model (SRM) proposed by *Wahba* [40.22] and *Engle* et al. [40.23]. Specifically, if $f_1 = \cdots = f_n \equiv f$ and $J = 1$, then the TW-SLM simplifies to the SRM, which has one nonparametric component and one finite-dimensional regression parameter. Much work has been done concerning the properties of the semiparametric least squares estimator (LSE) in the SRM, see for example, *Heckman* [40.24] and *Chen* [40.25]. It has been shown that, under appropriate regularity conditions, the semiparametric least squares estimator of the finite-dimensional parameter in the SRM is asymptotically normal, although the rate of convergence of the estimator of the nonparametric component is slower than $n^{1/2}$.

40.2 Semiparametric M-Estimation in TW-SLM

We describe two approaches of semiparametric M-estimation in the TW-SLM. The first one uses linear combinations of certain basis functions (e.g. B-splines) to approximate the normalization curves. The second one uses the local regression technique for estimation in the TW-SLM. Three important special cases in each approach include the least squares estimator, the least absolute deviation estimator, and Huber's robust estimator [40.26].

40.2.1 Basis-Based Method

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iJ})'$, $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iJ})'$ and $f(\boldsymbol{x}_i) \equiv [f(x_{i1}), \ldots, f(x_{iJ})]'$ for a univariate function f. We write the TW-SLM (40.1) in vector notation as

$$\boldsymbol{y}_i = \boldsymbol{\beta} z_i + f_i(\boldsymbol{x}_i) + \epsilon_i, \quad i = 1, \ldots, n. \tag{40.3}$$

Let $\Omega_0^{J\times d}$ be the space of all $J \times d$ matrices $\boldsymbol{\beta} \equiv (\beta_1, \ldots, \beta_J)'$ satisfying $\sum_{j=1}^J \beta_j = 0$. It is clear from the definition of the TW-SLM model (40.3) that $\boldsymbol{\beta}$ is identifiable only up to a member in $\Omega_0^{J\times d}$, since we may simply replace β_j by $\beta_j - \sum_{k=1}^J \beta_k/J$ and $f_i(x)$ by $f_i(x) + \sum_{k=1}^J \beta_k' z_i/J$ in (40.1). In what follows, we assume

$$\boldsymbol{\beta} \in \Omega_0^{J\times d} \equiv \left\{ \boldsymbol{\beta} : \sum_{j=1}^J \beta_j = 0 \right\}. \tag{40.4}$$

Let $b_{i1}, \ldots, b_{i,K_i}$ be K_i B-spline basis functions[40.27]. Let

$$S_i \equiv \overline{\{b_{i0}(x) \equiv 1, b_{ik}(x), k = 1, \ldots, K_i\}} \tag{40.5}$$

be the spaces of all linear combinations of the basis functions. We note that wavelet, Fourier and other types of basis functions can also be used. We approximate f_i

by

$$\alpha_{i0} + \sum_{k=1}^{K_i} b_{ik}(x)\alpha_{ik} \equiv \boldsymbol{b}_i(x)'\alpha_i, \in S_i$$

where $\boldsymbol{b}_i(x) = [1, b_{i1}(x), \ldots, b_{i,K_i}(x)]'$, and $\alpha_i = (\alpha_{i0}, \alpha_{i1}, \ldots, \alpha_{iK_i})'$ are coefficients to be estimated from the data. Let $\boldsymbol{b}f = (f_1, \ldots, f_n)$ and

$$M_s(\boldsymbol{\beta}, \boldsymbol{f}) = \sum_{i=1}^{n}\sum_{j=1}^{J} m_s\left[y_{ij} - f_i(x_{ij}) - \beta_j' z_i\right], \tag{40.6}$$

where m_s is an appropriate convex function which may also depend on a scale parameter s. Three important special cases are $m_s(t) = t^2$, $m_s(t) = |t|$, and the Huber ρ function. We define the semiparametric M-estimator of $\{\boldsymbol{\beta}, \boldsymbol{f}\}$ to be the $\{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{f}}\} \in \Omega_0^{J\times d} \times \prod_{i=1}^{n} S_i$ that minimizes $M_s(\boldsymbol{\beta}, \boldsymbol{f})$. It is often necessary to consider a scale parameter s in robust estimation. This scale parameter usually needs to be estimated jointly with $(\boldsymbol{\beta}, \boldsymbol{f})$.

One question is how to determine the number of basis functions K_i. For the purpose of normalization, it is reasonable to use the same K for all the arrays, that is, let $K_1 = \cdots = K_n \equiv K$. This will make normalization consistent across the arrays. For the cDNA microarray data, the total intensity has positive density over a finite interval, typically [0, 16]. For the cubic polynomial splines, we have used the number of knots $K = 12$, and the data percentiles as the knots in the R function bs.

40.2.2 Local Regression (Lowess) Method

We can also use the lowess method [40.18] for the estimation of TW-SLM. Let W_λ be a kernel function with window width λ. Let

$$s_p(t; \boldsymbol{\alpha}, x) = \alpha_0(x) + \alpha_1(x)t + \cdots + \alpha_p(x)t^p$$

be a polynomial in t with order p, where $p = 1$ or 2 are common choices. The objective function of the *lowess* method for the TW-SLM is

$$M_s(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{n}\sum_{j=1}^{J}\sum_{k=1}^{J} W_\lambda(x_{ik}, x_{ij}) m_s\Big[y_{ik} - s_p(x_{ik}; \boldsymbol{\alpha}, x_{ij}) - z_i'\beta_k\Big]. \tag{40.7}$$

Let $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})$ be the value that minimizes M_L. The lowess M-estimator of f_i at x_{ij} is $\hat{f}_i(x_{ij}) = s_p(x_{ij}, \hat{\boldsymbol{\alpha}}, x_{ij})$.

40.2.3 Back-Fitting Algorithm in TW-SLM

In both the basis-based and local regression methods, we use a back-fitting algorithm [40.28] to compute the semiparametric M-estimators. For the M-estimator based on the basis spaces S_i defined in (40.6), set $\boldsymbol{\beta}^{(0)} = 0$. For $k = 0, 1, 2, \ldots,$

- Step 1: compute $\boldsymbol{f}^{(k)}$ by minimizing $M_s\left(\boldsymbol{f}, \boldsymbol{\beta}^{(k)}\right)$ with respect to the space $\prod_{i=1}^{n} S_i$.
- Step 2: for the $\boldsymbol{f}^{(k)}$ computed above, obtain $\boldsymbol{\beta}^{(k+1)}$ by minimizing $M_s\left(\boldsymbol{f}^{(k)}, \boldsymbol{\beta}\right)$ with respect to $\boldsymbol{\beta}$ in $\Omega_0^{J\times d}$.

Iterate between steps 1 and 2 until the desired convergence criterion is satisfied.

For strictly convex m, e.g., $m(t) = t^2$ or $m(t) = |t|$, the algorithm converges to the unique global optimal point. The back-fitting algorithm can be also applied to the lowess M-estimators. When $m(t) = t^2$, then computation consists of a series of weighted regression problems.

40.2.4 Semiparametric Least Squares Estimation in TW-SLM

An important special case of the M-estimator is the least squares (LS) estimator, which has an explicit form in the TW-SLM [40.19, 20]. The LS objective function is

$$D^2(\boldsymbol{\beta}, \boldsymbol{f}) = \sum_{i=1}^{n}\sum_{j=1}^{J}\left[y_{ij} - f_i(x_{ij}) - z_i'\beta_j\right]^2.$$

The semiparametric least squares estimator (SLSE) of $\{\boldsymbol{\beta}, \boldsymbol{f}\}$ is the $\{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{f}}\} \in \Omega_0^{J\times d} \times \prod_{i=1}^{n} S_i$ that minimizes $D^2(\boldsymbol{\beta}, \boldsymbol{f})$. That is,

$$\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{f}}\right) = \arg\min_{(\boldsymbol{\beta}, \boldsymbol{f}) \in \Omega_0^{J\times d} \times \prod_{i=1}^{n} S_i} D^2(\boldsymbol{\beta}, \boldsymbol{f}). \tag{40.8}$$

Denote the spline basis matrix for the i-th array by

$$B_i = \begin{pmatrix} B_{i1}' \\ \vdots \\ B_{iJ}' \end{pmatrix} = \begin{pmatrix} 1 & b_{i1}(x_{i1}) & \ldots & b_{iK_i}(x_{i1}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & b_{i1}(x_{iJ}) & \ldots & b_{iK_i}(x_{iJ}) \end{pmatrix}.$$

Define the projection matrix Q_i as

$$Q_i = B_i\left(B_i'B_i\right)^{-1}B_i', \quad i = 1, \ldots, n.$$

Let $\alpha_i = (\alpha_{i0}, \ldots, \alpha_{iK_i})'$ be the spline coefficients for the estimation of f_i and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)'$. We can write $D^2(\boldsymbol{\beta}, \boldsymbol{\alpha}) = D^2(\boldsymbol{\beta}, \boldsymbol{f})$. Then the problem of minimizing

$D^2(\boldsymbol{\beta}, \boldsymbol{\alpha})$ with respect to $(\boldsymbol{\beta}, \boldsymbol{\alpha})$ is equivalent to solving the linear equations:

$$\hat{\boldsymbol{\beta}} \sum_{i=1}^{n} z_i z_i' + \sum_{i=1}^{n} B_i \hat{\alpha}_i z_i' = \sum_{i=1}^{n} \boldsymbol{y}_i z_i', \; B_i' B_i \hat{\alpha}_i + B_i' \hat{\boldsymbol{\beta}} z_i = B_i' \boldsymbol{y}_i \; .$$

Let $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}})$ be the solution. We define $\hat{f}_i(x) \equiv \boldsymbol{b}_i(x)' \hat{\alpha}_i$, $i = 1, \ldots, n$.

Using (40.3), it can be shown that the SLSE (40.8) equals

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\| \boldsymbol{y}_i - (I_J - Q_i)\, \boldsymbol{\beta} z_i \right\|^2 . \quad (40.9)$$

In the special case when $d = 1$ (scalar β_j) and $\boldsymbol{\beta}$ is a vector in $\mathbb{R}^J$, (40.9) is explicitly

$$\hat{\boldsymbol{\beta}} = \hat{\Lambda}^{-1} \left[\frac{1}{n} \sum_{i=1}^{n} (I_J - Q_i) \boldsymbol{y}_i z_i' \right] , \quad (40.10)$$

since $I_J - Q_i$ are projections in $\mathbb{R}^J$, where $z_i = 1$ (scalar) and, where

$$\hat{\Lambda}_{J,n} \equiv \frac{1}{n} \sum_{i=1}^{n} (I_J - Q_i) \otimes z_i z_i' \; . \quad (40.11)$$

We note that $\hat{\Lambda}$ can be considered as the observed information matrix. Here and below, $\boldsymbol{A}^{-1}$ denotes the generalized inverse of matrix $\boldsymbol{A}$, defined by $\boldsymbol{A}^{-1}\boldsymbol{x} \equiv \arg\min\left(\|\boldsymbol{b}\| : \boldsymbol{A}\boldsymbol{b} = \boldsymbol{x} \right)$. If $\boldsymbol{A}$ is a symmetric matrix with eigenvalues λ_j and eigenvectors $\boldsymbol{v}_j$, then $\boldsymbol{A} = \sum_j \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j'$ and $\boldsymbol{A}^{-1} = \sum_{\lambda_j \neq 0} \lambda_j^{-1} \boldsymbol{v}_j \boldsymbol{v}_j'$.

For general z_i and $d \geq 1$, (40.9) is still given by (40.10) with

$$\hat{\Lambda}_{J,n} \equiv \frac{1}{n} \sum_{i=1}^{n} (I_J - Q_i) \otimes z_i z_i' \; . \quad (40.12)$$

The information operator (40.11) is an average of tensor products, i. e. a linear mapping from $\Omega_0^{J \times d}$ to $\Omega_0^{J \times d}$ defined by $\hat{\Lambda}\boldsymbol{\beta} \equiv n^{-1} \sum_{i=1}^{n} (I_J - Q_i) \boldsymbol{\beta} z_i z_i'$.

Although the SLSE has an explicit expression, direct computation of SLSE involves inversion of a large $J \times J$ matrix. So we use the back-fitting algorithm. In this case, computation in each step of the back-fitting algorithm becomes an easier least squares problem and has explicit expressions as follows. Set $\boldsymbol{\beta}^{(0)} = \boldsymbol{0}$. For $k = 0, 1, 2, \ldots,$

- Step 1: compute $\boldsymbol{\alpha}^{(k)}$ by minimizing $D^2(\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$. The explicit solution is

$$\alpha_i^{(k)} = \left(B_i' B_i \right)^{-1} B_i' \left(\boldsymbol{y}_i - \boldsymbol{\beta}^{(k)} z_i \right), \; i = 1, \ldots, n \; .$$

- Step 2: given the $\boldsymbol{\alpha}^{(k)}$ computed in step 1, let $f_i^{(k)}(x) = \boldsymbol{b}_i(x)' \alpha_i^{(k)}$, compute $\boldsymbol{\beta}^{(k+1)}$ by minimizing $D_w\left(\boldsymbol{\beta}, \boldsymbol{\alpha}^{(k)}\right)$ with respect to $\boldsymbol{\beta}$. The explicit solution is

$$\hat{\beta}_j^{(k+1)} = \left(\sum_{i=1}^{n} z_i z_i' \right)^{-1} \sum_{i=1}^{n} z_i \left[y_{ij} - f_i^{(k)}\left(x_{ij}\right) \right] , \quad j = 1, \ldots, J \; . \quad (40.13)$$

The algorithm converges to the sum of residual squares. Suppose that the algorithm meets the convergence criterion at step K. Then the estimated values of β_j are $\widehat{\beta}_j = \beta_j^{(K)}$, $j = 1, \ldots, J$, and the estimated normalization curves are

$$\hat{f}_i(x) = \boldsymbol{b}_i(x)' \alpha_i^{(K)} = \boldsymbol{b}_i(x)' (B_i' B_i)^{-1} B_i' (\boldsymbol{y}_i - \hat{\boldsymbol{\beta}} z_i) \, , \quad i = 1, \ldots, n \; . \quad (40.14)$$

The algorithm described above can be conveniently implemented in the statistical computing environment R [40.29]. Specifically, steps 1 and 2 can be solved by the function LM in R. The function BS can be used to create a basis matrix for the polynomial splines.

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iJ})'$ and $f_i(\boldsymbol{x}_i) = [f_i(x_{i1}), \ldots, f_i(x_{iJ})]'$. Let $Q_i = B_i (B_i' B_i)^{-1} B_i'$. By (40.14), the estimator of $f_i(\boldsymbol{x}_i)$ is

$$\hat{f}_i(\boldsymbol{x}_i) = Q_i (\boldsymbol{y}_i - \hat{\boldsymbol{\beta}} z_i) \, .$$

Thus the normalization curve is the result of the linear smoother Q_i operating on $\boldsymbol{y}_i - \hat{\boldsymbol{\beta}} z_i$. The gene effect $\hat{\boldsymbol{\beta}} z_i$ is removed from $\boldsymbol{y}_i$. In comparison, the lowess normalization method does not remove the gene effect. An analogue of the lowess normalization, but using polynomial splines, is

$$\tilde{f}_i(\boldsymbol{x}_i) = Q_i \boldsymbol{y}_i = B_i \boldsymbol{\alpha}_i^{(0)} \, . \quad (40.15)$$

We shall call (40.15) a *spline* normalization method. Comparing $\hat{f}_i(\boldsymbol{x}_i)$ with $\tilde{f}_i(\boldsymbol{x}_i)$, we find that, if there is a relatively large percentage of differentially expressed genes, the difference between these two normalization curves can be large. The magnitude of the difference also depends on the magnitude of the gene effects.

40.3 Extensions of the TW-SLM

In this section, we describe three models that are extensions of the basic TW-SLM. These models include the multi-way SLM (MW-SLM); a model that incorporates control genes in the normalization; and a model for simultaneous location and scale normalization.

40.3.1 Multi-Way Semilinear Models

Just as TW-SLM is a semilinear extension of two-way ANOVA, for data sets with a higher-dimensional structure, multi-way ANOVA can be extended to multi-way semilinear models (MW-SLM) in the same manner by including nonparametric and linear functions of covariates as the main and interactive terms/effects in the model. This connection between ANOVA and MW-SLM is important in design of experiments and in understanding and interpretation of the contribution of different effects and identifiability conditions. The examples below are motivated by real data sets.

In model (40.1), it is only made explicit that the normalization curve f_i is array-dependent. It is straightforward to construct a 3W-SLM to normalize the data at the printing-pin block level:

$$y_{ikj} = f_{ik}(x_{ikj}) + z_i'\beta_{kj} + \epsilon_{ikj}\,, \qquad (40.16)$$

with the identifiability condition $\sum_j \beta_{kj} = 0$, where y_{ikj} and x_{ikj} are the log-intensity ratio and log-intensity product of gene j in the k-th block of array i, respectively. Model (40.16) includes nonparametric components for the block and array effects and their interaction and linear components for the gene effects and their interaction with the block effects. It was used in *Huang* et al. [40.21] to analyze the Apo A1 data [40.7], as an application of the TW-SLM (for each fixed k) at the block level. The interaction between gene and block effects is present in (40.16) since we assume that different sets of genes are printed in different blocks. If a replication of the same (or entire) set of genes is printed in each block, we may assume no interaction between gene and block effects ($\beta_{kj} = \beta_j$) in (40.16) and reduce it to the TW-SLM with (i, k) as a single index, treating a block/array in (40.16) as an array in (40.1).

As an alternative to (40.16) we may also use constants to model the interaction between array and block effects as in ANOVA, resulting in the model

$$y_{ikj} = f_i(x_{ikj}) + \gamma_{ik} + z_i'\beta_{kj} + \epsilon_{ikj}\,, \qquad (40.17)$$

with identifiability conditions $\sum_i \gamma_{ik} = \sum_k \gamma_{ik} = 0$ and $\sum_{kj} \beta_{kj} = 0$. This can be viewed as an extension of the three-way ANOVA model $Ey_{ikj} = \mu + \alpha_{i\bullet\bullet} + \gamma_{ik\bullet} + \beta_{\bullet k\bullet} + \beta_{\bullet kj} + \beta_{\bullet\bullet j}$ without $\{i, j\}$ and three-way interactions, via $\mu + \alpha_{i\bullet\bullet} \Rightarrow f_i$ and $\beta_{\bullet k\bullet} + \beta_{\bullet kj} + \beta_{\bullet\bullet j} \Rightarrow \beta_{kj}$. Note that the main block effects are represented by f_{ik} in (40.16) and by β_{kj} in (40.17).

Our approach easily accommodates designs where genes are printed multiple times in each array. Such a design is helpful for improving the precision and for assessing the quality of an array using the coefficient of variation [40.12]. Suppose there is a matrix of printing-pin blocks in each array and that a replication of the same (or entire) set of genes is printed in each column of blocks in the matrix in each array. As in (40.17), a 4W-SLM can be written as

$$y_{icrj} = f_i(x_{icrj}) + \gamma_{icr} + z_i'\beta_{rj} + \epsilon_{icrj} \qquad (40.18)$$

for observations with the j-th gene in the block at c-th column and r-th row of the matrix in the i-th array, with identifiability conditions $\sum_i \gamma_{icr} = \sum_r \gamma_{icr} = 0$ and $\sum_{rj} \beta_{rj} = 0$, with or without the three-way interaction or the interaction between the column and row effects in γ_{icr}. Note that the matrix of blocks does not have to match the physical columns and rows of printing-pin blocks. In model (40.18), the only nonparametric component is the array effects and the block effects are modeled as in ANOVA. If the block effects also depend on the log-intensity product x_{icrj}, the f_i and γ_{icr} in (40.18) can be combined into $f_{icr}(x_{icrj})$, resulting in the TW-SLM (for each fixed r) at the row level, which is equivalent to (40.16). If the replication of genes is not balanced, we may use a MW-SLM derived from an ANOVA model with incomplete/unbalanced design or the modeling methodologies described in Sect. 40.2.

From the above examples, it is clear that, in an MW-SLM, the combination of main and interactive effects represented by a term is determined by the labeling of the parameter (not that of the covariates) of the term as well as the presence or absence of associated identifiability conditions. Furthermore, since the center of a nonparametric component, e.g. $\sum_j f_i(x_{ij})$ in a TW-SLM, is harder to interpret, identifiability conditions are usually imposed on parametric components. As a result, a nonparametric component representing an interactive effect usually represents all the associated main effects as well, and many MW-SLMs are equivalent to an implementation of the TW-SLM with a suitable partition of data, as in (40.16).

40.3.2 Spiked Genes and Incorporation of Prior Knowledge in the MW-SLM

We describe three methods to incorporate prior knowledge in an MW-SLM: augmenting models, coding covariates, and imposing linear constraints. An important application of these methods is inclusion of spiked genes in normalization.

In many customized microarray experiments, it is possible to include a set of spiked genes with equal concentrations in the Cy5 and Cy3 channels. An important reason to use spiked genes is to calibrate scanning parameters, for example, intensity levels from the spiked genes can be used for tuning the laser power in each scanning channel in order to balance the Cy5 and Cy3 intensities. Spiked genes do not necessarily show an observed 1:1 ratio due to experimental variations. Because the number of spiked genes is often small, it is not adequate just to use the spiked genes as the basis for normalization.

Let y_{ik}^s and x_{ik}^s be the log-intensity ratio and product of the k-th spiked gene in the i-th array, $i = 1, \ldots, n$, $k = 1, \ldots, K$. Then we can augment the TW-SLM (40.1) as follows:

$$y_{ik}^s = f_i\left(x_{ik}^s\right) + \varepsilon_{ik}^s,\ y_{ij} = f_i\left(x_{ij}\right) + z_i'\beta_j + \varepsilon_{ij}\ . \tag{40.19}$$

The first equation is for the spiked genes, whose corresponding β_k^s are zero. Since a common f_i is used in (40.19) for each array, data from both spiked genes and genes under study contribute to the estimation of normalization curves as well as gene effects. Note that the identifiability condition $\sum_j \beta_j = 0$ in (40.1) is neither necessary nor appropriate for (40.19).

We may also use the general TW-SLM (40.2) to model spiked genes by simply setting $z_{ij} = 0$ if a spiked gene is printed at the j-th spot in the i-th array and $z_{ij} = z_i$ otherwise, where z_i are the design variable for the i-th array as in (40.1).

A more general (but not necessarily simpler) method of incorporating prior knowledge is to impose constraints in addition or as alternatives to the identifiability conditions in an MW-SLM. For example, we set $\beta_j = 0$ if j corresponds to a spiked gene, and $\beta_{j_1} = \cdots = \beta_{j_r}$ if there are r replications of a experimental gene at spots $\{j_1, \ldots, j_r\}$ in each array.

40.3.3 Location and Scale Normalization

The models we described above are for location normalization. It is often necessary to perform scale normalization to make arrays comparable in scale. The standard approach is to perform scale normalization after the location normalization, as discussed in *Yang* et al. [40.11], so that normalization is completed in two separate steps. We can extend the MW-SLM to incorporate the scale normalization by introducing a vector of array-specific scale parameters $(\tau_1, \ldots, \tau_n)$, as in

$$\frac{y_{ij} - f_i(x_{ij})}{\tau_i} = z_i'\beta_j + \varepsilon_{ij}, \quad i = 1, \ldots, n\ , \quad j = 1, \ldots, J\ , \tag{40.20}$$

for the TW-SLM, where $\tau_1 \equiv 1$ and the τ_i are restricted to be strictly positive. A more general model would allow τ_i also to depend on the total intensity levels.

40.4 Variance Estimation and Inference for β

In addition to being a standalone model for normalization, the TW-SLM can also be used for detecting differentially expressed genes. For this purpose, we need to estimate the variance of $\hat{\boldsymbol{\beta}}$. This requires the estimation of residual variances.

We have considered the model in which the residual variances depend smoothly on the total intensity values, and such dependence may vary from array to array [40.21]. The model is

$$\sigma_{ij}^2 = \sigma_i^2(x_{ij}),\ i = 1, \ldots, n,\ j = 1, \ldots, J\ ,$$

where σ_i^2 is a smooth positive function. This model takes into account the possible array-to-array variations in the variances. Because of the smoothness assumption on σ_i^2, this model says that, in each array, the genes with similar expression intensity values also have similar residual variances. This is a reasonable assumption, since for many microarray data, the variability of the log-intensity ratio depends on the total intensity. In particular, it is often the case that the variability is higher in the lower range of the total intensity than in the higher range.

We use the method proposed by *Ruppert* et al. [40.30] and *Fan* and *Yao* [40.31] in estimating the variance function in a nonparametric regression model. For each $i = 1, \ldots, n$, we fit a smooth curve through the scatter plot $(x_{ij}, \hat{\epsilon}_{ij}^2)$, where $\hat{\epsilon}_{ij}^2 = (y_{ij} - \hat{f}_i(x_{ij}) - z_i'\hat{\beta}_j)^2$. This is equivalent to fitting the nonparametric regression

model

$$\hat{\epsilon}_{ij}^2 = \sigma_i^2(x_{ij}) + \tau_{ij},\ \ j = 1, \ldots, J\ ,$$

for $i = 1, \ldots, n$, where τ_{ij} is the residual term in this model. We use the same spline bases as in the estimation of f_i (40.14). The resulting spline estimator $\hat{\sigma}_i^2$ can be expressed as

$$\hat{\sigma}_i^2(x) = \boldsymbol{b}_i'(x)(B_i'B_i)^{-1}B_i'\hat{\boldsymbol{\epsilon}}_i^2, \tag{40.21}$$

where $\hat{\boldsymbol{\epsilon}}_i^2 = (\hat{\epsilon}_{i1}^2, \ldots, \hat{\epsilon}_{iJ}^2)'$. The estimator of σ_{ij}^2 is then $\hat{\sigma}_{ij}^2 = \hat{\sigma}_i^2(x_{ij})$.

We can now approximate the variance of $\hat{\beta}_j$ as follows [40.21]. Let $Z_n = \sum_{i=1}^n z_i z_i'$. Based on (40.13), we have

$$\begin{aligned}\mathrm{var}(\hat{\beta}_j) &\approx Z_n^{-1}\left[\sum_{i=1}^n z_i z_i' \mathrm{var}(\epsilon_{ij})\right] Z_n^{-1} \\ &\quad + Z_n^{-1}\left[\sum_{i=1}^n z_i z_i' \mathrm{var}\left[\hat{f}_i(x_{ij})\right]\right] Z_n^{-1} \\ &\equiv \Sigma_{\epsilon,j} + \Sigma_{f,j}\ .\end{aligned}$$

The variance of $\hat{\beta}_j$ consists of two components. The first component represents the variation due to the residual errors in the TW-SLM, and the second component is due to the variation in the estimated normalization curves.

For the first term $\Sigma_{\epsilon,j}$, we have

$$\Sigma_{\epsilon,j} = Z_n^{-1}\left(\sum_{i=1}^n z_i z_i' \sigma_{ij}^2\right) Z_n^{-1}\ .$$

Suppose that $\hat{\sigma}_{ij}^2$ is a consistent estimator of σ_{ij}^2, which will be given below. We estimate $\Sigma_{\epsilon,j}$ by

$$\hat{\Sigma}_{\epsilon,j} = Z_n^{-1}\left(\sum_{i=1}^n z_i z_i' \hat{\sigma}_{ij}^2\right) Z_n^{-1}\ .$$

For the second term $\Sigma_{f,j}$, we approximate $\hat{\boldsymbol{f}}_i$ by the ideal normalization curve, that is,

$$\begin{aligned}\hat{f}_i(\boldsymbol{x}_i) &= Q_i\left(\boldsymbol{y}_i - \hat{\boldsymbol{\beta}} z_i\right) \approx Q_i\left(\boldsymbol{y}_i - \boldsymbol{\beta} z_i\right) \\ &= Q_i[\boldsymbol{\epsilon}_i + f_i(\boldsymbol{x}_i)]\ .\end{aligned}$$

Therefore, conditional on $\boldsymbol{x}_i$, we have,

$$\mathrm{var}\left[\hat{f}_i(\boldsymbol{x}_i)\right] \approx Q_i \mathrm{var}(\boldsymbol{\epsilon}_i) Q_i\ ,$$

and

$$\mathrm{var}\left[\hat{f}_i(x_{ij})\right] = \boldsymbol{e}_j' Q_i \mathrm{var}(\boldsymbol{\epsilon}_i) Q_i \boldsymbol{e}_j\ ,$$

where $\boldsymbol{e}_j$ is the unit vector whose j-th element is 1. Let $\hat{\Sigma}_i$ be an estimator of $\mathrm{var}(\boldsymbol{\epsilon}_i)$. We estimate $\Sigma_{f,j}$ by

$$\hat{\Sigma}_{f,j} = Z_n^{-1}\boldsymbol{e}_j'\left(\sum_{i=1}^n Q_i \hat{\Sigma}_i Q_i\right)\boldsymbol{e}_j Z_n^{-1}\ .$$

Finally, we estimate $\mathrm{var}\left(\hat{\beta}_j\right)$ by

$$\hat{\Sigma}_{\beta,j} = \hat{\Sigma}_{\epsilon,j} + \hat{\Sigma}_{f,j}\ . \tag{40.22}$$

Then a test for the contrast $c'\beta_j$, where c is a known contrast vector, is based on the statistic

$$t_j = \frac{c'\hat{\beta}_j}{\sqrt{c'\hat{\Sigma}_{\beta,j}c}}\ .$$

As is shown in Sect. 40.6, for large J, the distribution of t_j can be approximated by the standard normal distribution under the null $c'\beta_j = 0$. However, to be conservative, we use a t distribution with an appropriate number of degrees of freedom to approximate the null distribution of t_j when $c'\beta_j = 0$. For example, for a direct comparison design, the number of degrees of freedom is $n-1$. For a reference design in a two sample comparison, the variances for the two groups can be estimated separately, and then Welch's correction for the degrees of freedom can be used. Resampling methods such as permutation or bootstrap can also be used to evaluate the distribution of t_j.

Another approach is to estimate σ_{ij}^2 jointly with $(\boldsymbol{f}, \boldsymbol{\beta})$. This approach is computationally more intensive but may yield more efficient estimates of $(\boldsymbol{\beta}, \boldsymbol{f})$ and σ_{ij}^2. Consider an approximation to σ_{ij}^2 using the spline basis functions:

$$\sigma_{ij}^2 = \sigma_i^2(x_{ij}) = \sum_{k=1}^{K_i} \gamma_{ik} b_k(x_{ij})\ . \tag{40.23}$$

Let $\boldsymbol{\gamma}$ be the collection of the γ_{ik}. Assuming normality for ε_{ij}, the negative likelihood function is

$$\ell(\boldsymbol{\beta}, \boldsymbol{f}, \boldsymbol{\gamma}) = -\prod_{i=1}^n \prod_{j=1}^J \frac{1}{\sigma_{ij}}\phi\left(\frac{y_{ij} - f_i(x_{ij}) - \beta_j' z_i}{\sigma_{ij}}\right)\ , \tag{40.24}$$

where ϕ is the density of $N(0,1)$. For robust M-estimation, we define the M-estimation objective function as

$$M_s(\boldsymbol{\beta}, \boldsymbol{f}, \boldsymbol{\gamma}) = \sum_{i=1}^n \sum_{j=1}^J \sigma_{ij} m_s\left(\frac{y_{ij} - f_i(x_{ij}) - \beta_j' z_i}{\sigma_{ij}}\right)\ . \tag{40.25}$$

Again, we can use a back-fitting algorithm for computing the M-estimators, but with an extra step in each iteration for $\boldsymbol{\gamma}$.

40.5 An Example and Simulation Studies

40.5.1 Apo A1 Data

We now illustrate the TW-SLM for microarray data by the Apo A1 data set of *Callow* et al. [40.7]. The analysis described here is from *Huang* et al. [40.21]. The purpose of this experiment is to identify differentially expressed genes in the livers of mice with very low high-density lipoprotein (HDL) cholesterol levels compared to inbred mice. The treatment group consists of eight mice with the apo A1 gene knocked out and the control group consists of eight C57BL/6 mice. For each of these mice, target cDNA is obtained from mRNA by reverse transcription and labeled using a red fluorescent dye (Cy5). The reference sample (green fluorescent dye Cy3) used in all hybridizations was obtained by pooling cDNA from the eight control mice. The target cDNA is hybridized to microarrays containing 5548 cDNA probes. This data set was analyzed by *Callow* et al. [40.7] and *Dudoit* et al. [40.32]. Their analysis uses lowess normalization and the two-sample t-statistic. Eight genes with multiple comparison adjusted permutation p-value ≤ 0.01 are identified.

We apply the proposed normalization and analysis method to this data set. As in *Dudoit* et al. [40.32], we use printing-tip-dependent normalization. The TW-SLM model used here is

$$y_{ikj} = f_{ik}(x_{ikj}) + z_i' \beta_{kj} + \varepsilon_{ikj} \,,$$

where $i = 1, \ldots, 16$, $k = 1, \ldots, 16$, and $j = 1, \ldots, 399$. Here i indexes arrays, k indexes printing-tip blocks, and j index genes in a block. ε_{ikj} are residuals with mean 0 and variance σ^2_{ikj}. We use the model

$$\sigma^2_{ikj} = \sigma^2_{ik}(x_{ikj}) \,,$$

where σ^2_{ik} are unknown smooth functions. We apply the printing-pin-dependent normalization and estimation approach described in Sect. 40.3.2. The covariate $z_i = (1, 0)'$ for the treatment group (apo A1 knock-out mice) and $z_i = (0, 1)'$ for the control group (C57BL/6 mice). The coefficient $\beta_{kj} = (\beta_{kj1}, \beta_{kj2})$. The contrast $\beta_{kj1} - \beta_{kj2}$ measures the expression difference for the j-th gene in the k-th block between the two groups.

To compare the proposed method with the existing ones, we also analyzed the data using the lowess normalization method as in *Dudoit* et al. [40.32], and a lowess-like method where, instead of using local regression, splines are used in estimating the normalization curves described in (40.15) at the end of Sect. 40.2. We refer to this method as the spline (normalization) method below.

As examples of the normalization results, Fig. 40.1 displays the M–A plots and printing-tip-dependent normalization curves in the 16 printing-pin blocks of the array from one knock-out mouse. The solid line is the normalization curve based on the TW-SLM model, and the dashed line is the lowess normalization curve. The degrees of freedom used in the spline basis function in the TW-SLM normalization is 12, and following *Dudoit* et al. [40.32], the span used in the lowess normalization is 0.40. We see that there are differences between the normalization curves based on the two methods. The lowess normalization curve attempts to fit each individual M–A scatter plot, without taking into account the gene effects. In comparison, the TW-SLM normalization curves do not follow the plot as closely as the lowess normalization. The normalization curves estimated using the spline method with exactly the same basis functions used in the TW-SLM closely resemble those estimated using the lowess method. Because they are indistinguishable by eye, these curves are not included in the plots.

Figure 40.2 displays the volcano plots of $-\log_{10}$ p-values versus the mean differences of log-expression values between the knock-out and control groups. In the first (left panel) volcano plot, both the normalization and estimation of $\boldsymbol{\beta}$ are based on the TW-SLM. We estimated the variances for $\hat{\beta}_{kj1}$ and $\hat{\beta}_{kj2}$ separately. These variances are estimated based on (40.21), which assumes that the residual variances depend smoothly on the total log-intensities. We then used Welch's correction for the degrees of freedom in calculating the p-values. The second (middle panel) plot is based on the lowess normalization method and use the two-sample t-statistics as in *Dudoit* et al. [40.32], but the p-values are obtained based on Welch's correction for the degrees of freedom. The third (right panel) plot is based on the spline normalization method and uses the same two-sample t-statistics as in the lowess method. The eight solid circles in the lowess volcano plot are the significant genes that were identified by *Dudoit* et al. [40.32]. These eight genes are also plotted as solid circles in the TW-SLM and spline volcano plots, and are significant based on the TW-SLM and spline methods, as can be seen from the volcano plots. Comparing the three volcano plots, we see that: (i) the $-\log_{10}$ p-values based on the TW-SLM method tend to be higher than those based on the lowess and

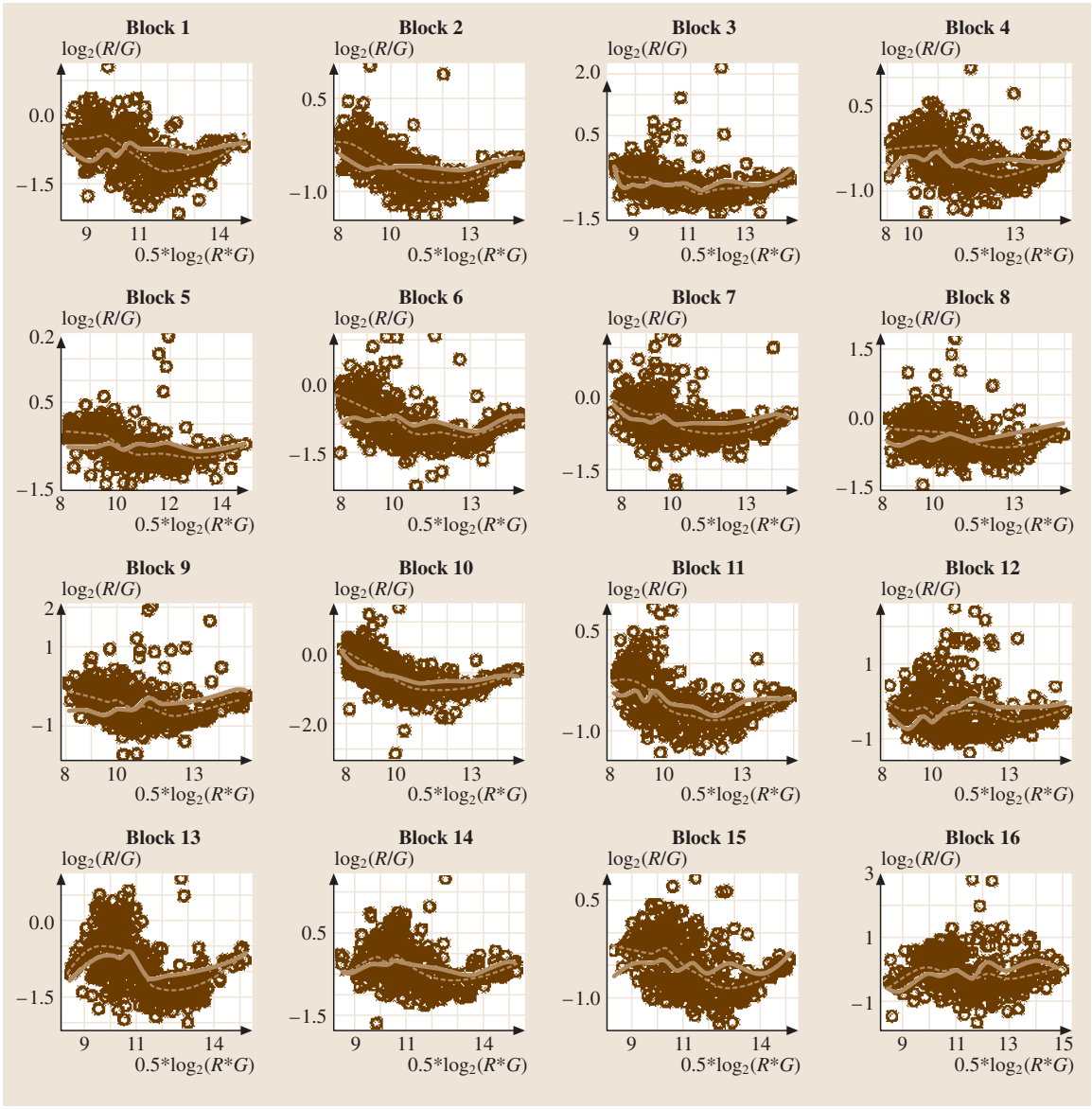


Fig. 40.1 Apo AI data: comparison of normalization curves in the 16 blocks of the array from one knock-out mouse in the treatment group. *Solid line*: normalization curve based on TW-SLM; *dashed line*: normalization curve based on lowess

spline methods; (ii) the p-values based on the lowess and spline methods are comparable.

Because we use exactly the same smoothing procedure in the TW-SLM and spline methods, and because the results between the lowess and spline methods are very similar, we conclude that the differences between the TW-SLM and lowess volcano plots are mostly due to the different normalization methods and two difference approaches for estimating the variances. We first examine the differences between the TW-SLM normalization values and the lowess as well as the spline normalization values. We plot the three pairwise scatter plots of estimated mean expression differences based on the TW-SLM, lowess, and spline normalization methods, see

Fig. 40.3. In each scatter plot, the solid line is the fitted linear regression line. For the TW-SLM versus lowess comparison (left panel), the fitted regression line is

$$y = 0.00029 + 1.090x\,. \tag{40.26}$$

The standard error of the intercept is 0.0018, so the intercept is negligible. The standard error of the slope is 0.01. Therefore, on average, the mean expression differences based on the TW-SLM normalization method are about 10% higher than those based on the lowess normalization method. For the TW-SLM versus spline comparison (middle panel), the fitted regression line and the standard errors are virtually identical to (40.26) and its associated standard errors. For the spline versus lowess comparison (right panel), the fitted regression line is

$$y = 0.00027 + 1.00257x\,. \tag{40.27}$$

The standard error of the intercept is 0.000 25, and the standard of the slope is 0.0015. Therefore, the mean expression differences based on the lowess and spline normalization methods are essentially the same, as can also be seen from the scatter plot in the right panel in Fig. 40.3.

Figure 40.4 shows the histograms of the standard errors obtained based on intensity-dependent smoothing defined in (40.21) using the residuals from the TW-SLM normalization (top panel), and the standard errors calculated for individual genes using the lowess and spline methods (middle and bottom panels). The standard errors (SE) based on the individual genes have a relatively large range of variation, but the range of standard errors based on intensity-dependent smoothing shrinks towards the middle. The SEs based on the smoothing method are more tightly centered around the median value of about 0.13. Thus, the analysis based on the smooth estimate of the error variances is less susceptible to the problem of artificially small p-values resulting from random small standard errors calculated from individual genes.

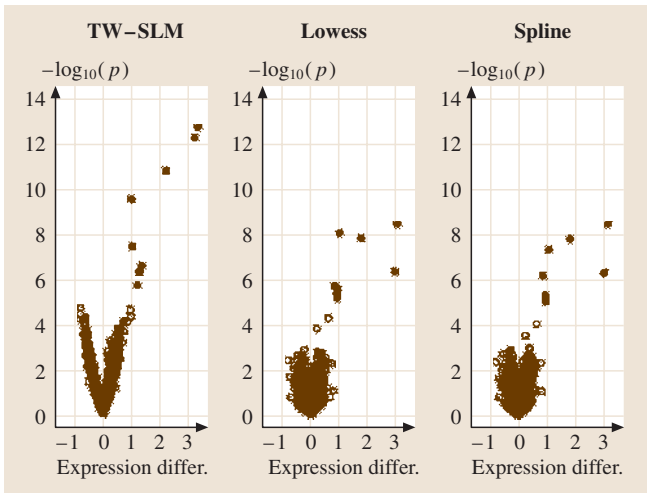


Fig. 40.2 Volcano plots: scatter plot of $-\log_{10}(p-\text{value})$ versus estimated mean expression value. The left panel shows the volcano plot based on the TW-SLM; the middle panel shows the plot based on the lowess method; the right panel shows the result based on the spline method

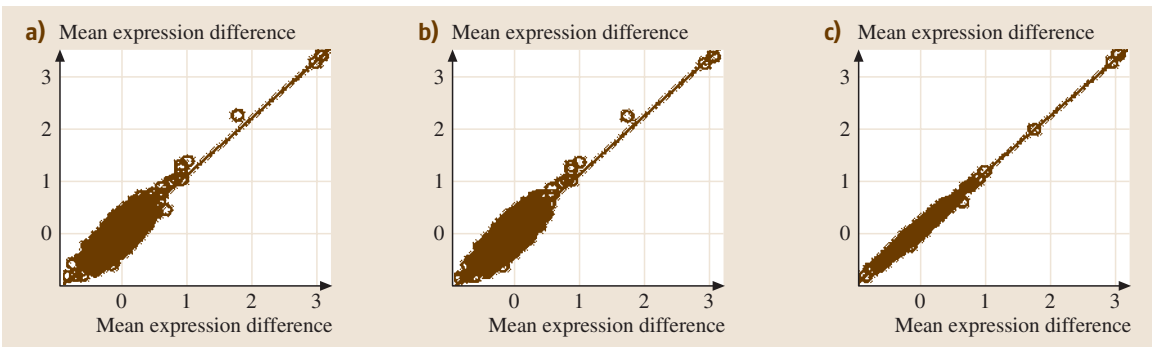


Fig. 40.3a–c Comparison of normalized expression values. *Left panel*: the scatter plot of normalized mean expression differences based on TW-SLM versus those based on lowess. *Middle panel*: The scatter plot of normalized mean expression differences based on TW-SLM versus those based on the spline method. *Right panel*: The scatter plot of normalized mean expression differences based on spline versus those based on lowess. **(a)** TW-SLM versus lowess, **(b)** TW-SLM versus spline, **(c)** spline versus lowess

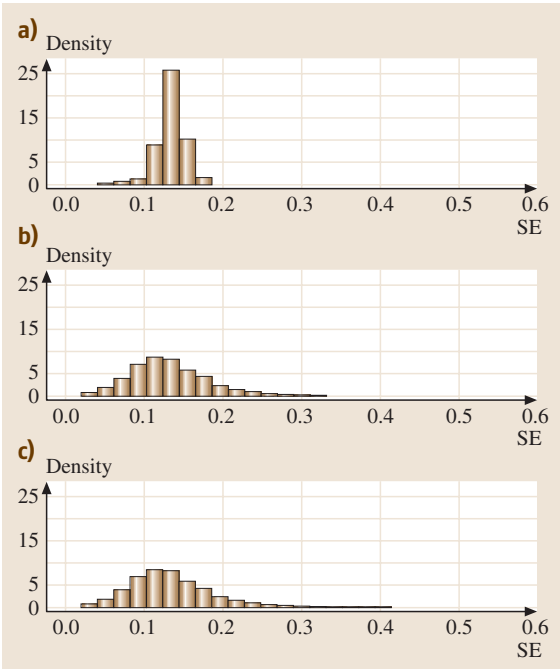


Fig. 40.4a–c Comparison of variance estimation methods. *Top panel*: The histogram of SE estimated based on smoothing as described in Sect. 40.3.2. *Middle panel*: SE estimated based on individual genes using the lowess method. *Bottom panel*: SE estimated based on individual genes using the spline method. **(a)** TW-SLM: SE based on smoothing **(b)** lowess: SE based on individual genes **(c)** spline: SE based on individual genes

40.5.2 Simulation Studies

We use simulation to compare the TW-SLM, lowess, and spline normalization methods with regard to the mean square errors (MSE) in estimating expression levels β_j. The simulation models and results are from *Huang* et al. [40.21]. Let α_1 and α_2 be the percentages of up- and down-regulated genes, respectively, and let $\alpha = \alpha_1 + \alpha_2$. We consider four models in our simulation.

- Model 1: there is no dye bias. So the true normalization curve is set at the horizontal line at 0. That is $f_i(x) \equiv 0, 1 \le i \le n$. In addition, the expression levels of up- and down-regulated genes are symmetric and $\alpha_1 = \alpha_2$.
- Model 2: as in model 1, the true normalization curves $f_i(x) \equiv 0, 1 \le i \le n$, but the percentages of up- and down-regulated genes are different. We set $\alpha_1 = 3\alpha_2$.
- Model 3: there are nonlinear and intensity-dependent dye biases. The expression levels of up- and down-regulated genes are symmetric and $\alpha_1 = \alpha_2$.
- Model 4: there is nonlinear and intensity-dependent dye bias. The percentages of up- and down-regulated genes are different. We set $\alpha_1 = 3\alpha_2$.

Models 1 and 2 can be considered as the baseline ideal case in which there is no channel bias. The data-generating process is as follows:

- Generate β_j. For most of the genes, we simulate $\beta_j \sim N(0, \tau_j^2)$. The percentage of such genes is $1-\alpha$. For up-regulated genes, we simulate $\beta_j \sim N(\mu, \tau_{\mathrm{U}j}^2)$ where $\mu > 0$. For down-regulated genes, we simulate $\beta_j \sim N(-\mu, \tau_{\mathrm{D}j}^2)$. We use $\tau_j = 0.6$, $\mu = 2$, $\tau_{\mathrm{U}j} = \tau_{\mathrm{D}j} = 1$.
- Generate x_{ij}. We simulate $x_{ij} \sim 16 \times Beta(a, b)$, where $a = 1$, $b = 2.5$.
- Generate ϵ_{ij}. We simulate $\epsilon_{ij} \sim N(0, \sigma_{ij}^2)$, where $\sigma_{ij} = \sigma(x_{ij})$. Here $\sigma(x) = 0.3 * x^{-1/3}$. So the error variance is higher at lower intensity range than at higher intensity range.
- In models 1 and 2, the log-intensity ratios are computed as $y_{ij} = f_i(x_{ij}) + \beta_j + \epsilon_{ij}$.

In models 3 and 4, the log-intensity ratios are computed according to a printing-tip-dependent model with $y_{ij} = \beta_j + f_{ik(j)}(x_{ij}) + \varepsilon_{ij}$, where the function $k(j)$ indicates the printing-pin block. This is equivalent to the model used in the analysis of the Apo A1 data in Sect. 40.5.1, with $z_i = 1$ there. We use

$$f_{ik} = \frac{a_{ik1} x^2 \sin(x/\pi)}{1 + a_{ik2} x^2} ,$$

where a_{i1} and a_{i2} are generated independently from the uniform distribution U(0.6, 1.4). Thus the normalization curves vary from block to block within an array and between arrays.

The number of printing-pin blocks is 16, and in each block there are 400 spots. The number of arrays in each data set is 10. The number of replications for each simulation is 10. Based on these 10 replications, we calculate the bias, variance, and mean square error of estimated expression values relative to the generating values. In each of the four cases, we consider two levels of the percentage of differentially expressed genes: $\alpha = 0.01$ and 0.06.

Tables 40.1–40.4 present the summary statistics of the MSEs for estimating the relative expression levels β_j in the four models described above. In Table 40.1

Table 40.1 Simulation results for model 1. 10 000 × Summary of MSE. The true normalization curve is the horizontal line at 0. The expression levels of up- and down-regulated genes are symmetric: $\alpha_1 = \alpha_2$, where $\alpha_1 + \alpha_2 = \alpha$

		Min.	1st quartile	Median	Mean	3rd quartile	Max.
$\alpha = 0.01$	TW-SRM	3.74	51.59	75.08	88.88	106.20	4980.00
	Lowess	3.38	50.72	72.77	87.89	105.10	7546.00
	Splines	7.08	58.93	85.35	98.25	121.10	4703.00
$\alpha = 0.06$	TW-SRM	6.53	50.03	74.74	93.92	107.30	5120.00
	Lowess	9.09	50.89	73.93	91.87	106.10	6230.00
	Splines	8.95	61.34	89.03	105.60	126.10	6480.00

Table 40.2 Simulation results for model 2. 10 000 × Summary of MSE. The true normalization curve is the horizontal line at 0. But the percentages of up- and down-regulated genes are different: $\alpha_1 = 3\alpha_2$, where $\alpha_1 + \alpha_2 = \alpha$

		Min.	1st quartile	Median	Mean	3rd quartile	Max.
$\alpha = 0.01$	TW-SRM	5.36	58.04	71.01	83.17	102.50	1416.00
	Lowess	8.86	67.69	95.80	107.40	131.00	1747.00
	Splines	8.91	65.53	94.40	110.40	135.10	1704.00
$\alpha = 0.06$	TW-SRM	6.66	47.85	68.55	78.49	97.50	1850.40
	Lowess	6.45	59.54	87.08	99.00	123.90	1945.10
	Splines	6.74	59.23	86.58	98.67	123.30	1813.10

Table 40.3 Simulation results for model 3. 10 000 × Summary of MSE. There are nonlinear and intensity-dependent dye biases. The expression levels of up- and down-regulated genes are symmetric: $\alpha_1 = \alpha_2$, where $\alpha_1 + \alpha_2 = \alpha$

		Min.	1st quartile	Median	Mean	3rd quartile	Max.
$\alpha = 0.01$	TW-SRM	5.56	46.15	66.72	87.23	93.91	1898.00
	Lowess	6.71	51.07	74.23	88.79	107.50	3353.00
	Splines	5.90	53.83	76.91	88.64	108.60	1750.00
$\alpha = 0.06$	TW-SRM	6.64	57.26	85.79	102.80	126.40	2290.00
	Lowess	7.39	57.19	85.47	107.70	128.10	2570.00
	Splines	9.37	69.26	102.80	122.80	148.50	2230.00

Table 40.4 Simulation results for model 4. 10 000 × Summary of MSE. There are nonlinear and intensity-dependent dye biases. The percentages of up- and down-regulated genes are different: $\alpha_1 = 3\alpha_2$, where $\alpha_1 + \alpha_2 = \alpha$

		Min.	1st quartile	Median	Mean	3rd quartile	Max.
$\alpha = 0.01$	TW-SRM	5.89	51.26	74.53	85.89	107.20	2810.00
	Lowess	9.29	68.30	101.60	118.60	140.00	4088.00
	Splines	9.68	67.85	98.82	119.80	141.00	2465.00
$\alpha = 0.06$	TW-SRM	4.96	54.12	79.92	98.79	122.70	2130.00
	Lowess	6.49	71.54	113.90	130.90	169.50	2474.00
	Splines	5.77	65.46	107.57	128.40	171.60	1898.00

for simulation model 1, in which the true normalization curve is the horizontal line at 0 and the expression levels of up- and down-regulated genes are symmetric, the TW-SLM normalization tends to have slightly higher MSEs than the lowess method. The spline method has higher MSEs than both the TW-SLM and lowess methods. In Table 40.2, when there is no longer symmetry in the expression levels of up- and down-regulated genes, the TW-SLM method has smaller MSEs than both the lowess and spline methods. In Table 40.3 for simulation model 3, there is nonlinear intensity-dependent dye bias, but there is symmetry between the up- and down-regulated genes. The TW-SLM has comparable but slightly smaller MSEs than the lowess method. The spline method has higher MSEs than both the TW-SLM and lowess methods. In Table 40.4

for simulation model 4, there is nonlinear intensity-dependent dye bias, and the percentages of up- and down-regulated genes are different, the TW-SLM has considerably smaller MSEs. We have also examined biases and variances. There are only small differences in variances among the TW-SLM, lowess, and spline methods. However, the TW-SLM method generally has smaller biases.

40.6 Theoretical Results

In this section, we provide theoretical results concerning the distribution of $\hat{\boldsymbol{\beta}}$ and the rate of convergence for the normalization of f_i. The proofs can be found in *Huang* et al. [40.21]. Our results are derived under subsets of the following four conditions. We assume that the data from different arrays are independent, and impose conditions on the n individual arrays. Our conditions depend on n only through the uniformity requirements across the n arrays, so that all the theorems in this section hold in the case of fixed $n \geq 2$ as the number of genes $J \to \infty$ as well as the case of $(n, J) \to (\infty, \infty)$ with no constraint on the order of n in terms of J. In contrast, *Huang* and *Zhang* [40.20] focused on applications with large number of arrays. The results in this section hold for any basis functions b_{ik} in (40.5), e.g. spline, Fourier, or wavelet bases, as long as Q_i in (40.9) are projections from $\mathbb{R}^J$ to $\{f(\boldsymbol{x}_i) : f \in S_i\}$ with $Q_i \boldsymbol{e} = \boldsymbol{e}$, where $\boldsymbol{e} = (1, \ldots, 1)'$. Furthermore, with minor modifications in the proof, the results hold when Q_i are replaced by nonnegative definite smoothing matrices A_i with their largest eigenvalues bounded by a fixed constant, see [40.20, 21].

Condition I: *In (40.3), $\boldsymbol{x}_i$, $i = 1, \ldots, n$, are independent random vectors, and for each i $\left(x_{ij}, j \leq J\right)$ are exchangeable random variables. Furthermore, for each $i \leq n$, the space S_i in (40.5) depends on design variables $\left(\boldsymbol{x}_k, z_k, k \leq n\right)$ only through the values of $\boldsymbol{x}_i$ and $(z_k, k \leq n)$.*

The independence assumption follows from the independence of different arrays, which is satisfied in a typical microarray experiment. The exchangeability condition within individual arrays is reasonable if there is no prior knowledge about the total intensity of expression values of the genes under study. It holds when $\left(x_{ij}, j \leq J\right)$ are conditionally independent identically distributed (iid) variables given certain (unobservable random) parameters, including within-array iid $x_{ij} \sim G_i$ as a special case. The exchangeability condition also holds if $\left(x_{ij}, j \leq J\right)$ are sampled without replacement from a larger collection of variables.

Condition II: *The matrix $Z_n \equiv \sum_{i=1}^n z_i z_i'$ is of full rank with $\max_{i \leq n} z_i' Z_n^{-1} z_i \leq \kappa^* < 1$.*

Condition II is satisfied by common designs such as the reference and direct comparison designs. Since $\sum_{i=1}^n Z_n^{-1} z_i z_i' = I_d$, $\sum_{i=1}^n z_i' Z_n^{-1} z_i = d$. In balanced designs or orthogonal designs with replications, $Z_n \propto I_d$, n is a multiplier of d, and $z_i' Z_n^{-1} z_i = \kappa^* = d/n < 1$ for all $i \leq n$. In particular, (40.6) describes a balanced design with $d = 1$, so that condition II holds as long as $n \geq 2$.

Condition III: *For the projections Q_i in (40.9), $K_{J,n}^* \equiv \max_{i \leq n} E\left[\mathrm{tr}(Q_i) - 1\right] = O\left(J^{1/2}\right)$.*

An assumption on the maximum dimensions of the approximation spaces is usually required in nonparametric smoothing. Condition III assumes that the ranks of the projections Q_i are uniformly of the order $O\left(J^{1/2}\right)$ to avoid overfitting, and more important, to avoid colinearity between the approximation spaces for the estimation of $(f_i(\boldsymbol{x}_i), i \leq n)$ and the design variables for the estimation of $\boldsymbol{\beta}$. Clearly, $E\left[\mathrm{tr}(Q_i) - 1\right] \leq K_i$ for the K_i in (40.5).

Condition IV: $\rho_{J,n}^* \equiv \max_{i \leq n} E\left\| f_i(\boldsymbol{x}_i) - Q_i f_i(\boldsymbol{x}_i)\right\|^2 / (J-1) \to 0$.

Condition IV demands that the ranges of the projections Q_i be sufficiently large that the approximation errors for $f_i(\boldsymbol{x}_i)$ are uniformly $O(1)$ in an average sense. Although this is the weakest possible condition on Q_i for the consistent estimation of $f_i(\boldsymbol{x}_i)$, the combination of conditions III and IV does require careful selection of spaces S_i in (40.5) and certain condition on the tail probability of x_{ij}. See the two examples in *Huang* et al. [40.21] that illustrate this point.

40.6.1 Distribution of $\hat{\beta}$

We now describe the distribution of $\hat{\boldsymbol{\beta}}$ in (40.9) conditionally on all the covariates and provide an upper bound for the conditional bias of $\hat{\boldsymbol{\beta}}$.

Let $\hat{\Lambda}$ be the information operator in (40.12). Define

$$\tilde{\beta}_{J,n} = -\Pi_{J,n}\boldsymbol{\beta} + \hat{\Lambda}_{J,n}^{-1}\left[\frac{1}{n}\sum_{i=1}^{n}(I_J - Q_i) f_i(\boldsymbol{x}_i) z_i'\right], \tag{40.28}$$

where $\Pi_{J,n}$ is the projection to $\{\boldsymbol{b} \in \Omega_0^{J\times d} : \hat{\Lambda}_{J,n}\boldsymbol{b} = 0\}$. Define

$$V_{J,n} = \frac{1}{n}\sum_{i=1}^{n} V_i \otimes z_i z_i' ,$$
$$V_i = (I_J - Q_i)\mathrm{var}(\boldsymbol{\epsilon}_i)(I_J - Q_i) . \qquad (40.29)$$

Here $\hat{\Lambda}_{J,n}^{-1}$, the generalized inverse of $\hat{\Lambda}_{J,n}$, is uniquely defined as a one-to-one mapping from the range of $\hat{\Lambda}_{J,n}$ to the space $(I_J \otimes I_d - \Pi_{J,n})\Omega_0^{J\times d} = \{\boldsymbol{b} \in \Omega_0^{J\times d} : \Pi_{J,n}\boldsymbol{b} = 0\}$. For any $J\times b$ matrix $\boldsymbol{b}$, the matrix $B = \hat{\Lambda}_{J,n}^{-1}\boldsymbol{b}$ can be computed by the following recursion:

$$B^{(k+1)} \leftarrow n(\boldsymbol{b} - \Pi_{J,n}\boldsymbol{b})Z_n^{-1} + \sum_{i=1}^{n} Q_i B^{(k)} z_i z_i' Z_n^{-1} \qquad (40.30)$$

with the initialization $B^{(1)} = n(\boldsymbol{b} - \Pi_{J,n}\boldsymbol{b})Z_n^{-1}$ and $Z_n = \sum_{i=1}^{n} z_i z_i'$.

Theorem 40.1
Let $\hat{\boldsymbol{\beta}}_{J,n}$, $\hat{\Lambda}$ and $V_{J,n}$ be as in (40.9), (40.12) and (40.29) respectively. Suppose that given $\{\boldsymbol{x}_i, i \le n\}$, $\boldsymbol{\epsilon}_i$ are independent normal vectors. Then, conditionally on $\{\boldsymbol{x}_i, i \le n\}$,

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \sim N\left(\tilde{\boldsymbol{b}}_{J,n}, \frac{1}{n}\hat{\Lambda}_{J,n}^{-1} V_{J,n} \hat{\Lambda}_{J,n}^{-1}\right) . \qquad (40.31)$$

In particular, for all $\boldsymbol{b} \in \Omega_0^{J\times d}$, $\lim_{k\to\infty} B^{(k)} = \hat{\Lambda}^{-1}\boldsymbol{b}$ with the $B^{(k)}$ in (40.30), and

$$\sigma_{J,n}^2(\boldsymbol{b}) \equiv \mathrm{var}\left[\mathrm{tr}\left(\boldsymbol{b}'\hat{\boldsymbol{\beta}}\right)\middle|\{\boldsymbol{x}_i, i \le n\}\right]$$
$$= \frac{1}{n^2}\sum_{i=1}^{n} z_i'\left(\hat{\Lambda}^{-1}\boldsymbol{b}\right)' V_i \left(\hat{\Lambda}^{-1}\boldsymbol{b}\right) z_i . \qquad (40.32)$$

Our next theorem provides sufficient conditions under which the bias of $\hat{\boldsymbol{\beta}}$ is of smaller order than its standard error.

Theorem 40.2
Suppose conditions I to IV hold. If $c_{J,n}/\rho_{J,n}^* \to \infty$, then

$$\sup\left\{E\min\left(1, \frac{\mathrm{tr}^2\left(\boldsymbol{b}'\tilde{\boldsymbol{b}}_{J,n}\right)}{\mathrm{tr}\left(\boldsymbol{b}Z_n^{-1}\boldsymbol{b}'\right)c_{J,n}}\right) : \boldsymbol{b} \in \Omega_0^{J\times d}, \boldsymbol{b} \neq 0\right\} = O(1). \qquad (40.33)$$

In particular, if given $\{\boldsymbol{x}_i, i \le n\}$, $\boldsymbol{\epsilon}_i$ are independent normal vectors with $\mathrm{var}(\boldsymbol{\epsilon}_i) \ge \sigma_*^2 I_J$ for certain $\sigma_* > 0$, then

$$\sup_{\boldsymbol{b}\in\Omega_0^{J\times d}, \boldsymbol{b}\neq 0}\left\{\sup_{x\in\mathbb{R}}\left|P\left(\frac{\mathrm{tr}\left(\boldsymbol{b}'\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right)}{\sigma_{J,n}(\boldsymbol{b})} \le x\right) - \Phi(x)\right|\right\} = O(1) , \qquad (40.34)$$

where Φ is the cumulative distribution function for $N(0, 1)$.

This result states that, under conditions I to IV, appropriate linear combinations of $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, such as contrasts, have an approximate normal distribution with mean zero and the approximation is uniform over all linear combinations. Therefore, this result provides theoretical justification for inference procedures based on $\hat{\boldsymbol{\beta}}$, such as those described in Sect. 40.3. Without the normality condition, (40.29) is expected to hold under the Lindeberg condition as $(n, J) \to (\infty, \infty)$, even in the case $n = O(J)$ [for example $n = O(\log J)$]. We assume the normality here so that (40.29) holds for fixed n as well as large n.

40.6.2 Convergence Rates of Estimated Normalization Curves $\hat{f}_i$

Normalization is not only important in detecting differentially expressed genes, it is also a basic first step for other high-level analysis, including classification and cluster analysis. Thus, it is of interest in itself to study the behavior of the estimated normalization curves. Here we study the convergence rates of $\hat{f}_i$.

Since $\hat{f}_i(\boldsymbol{x}_i) = Q_i(\boldsymbol{y}_i - \hat{\boldsymbol{\beta}}z_i)$, it follows from (40.3) that

$$\hat{f}_i(\boldsymbol{x}_i) = Q_i\left[f_i(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i\right] - Q_i\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)z_i . \qquad (40.35)$$

Therefore, the convergence rates of $\|\hat{f}_i(\boldsymbol{x}_i) - f_i(\boldsymbol{x}_i)\|$ are bounded by the sums of the rates of $\|Q_i\left[f_i(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i\right] - f_i(\boldsymbol{x}_i)\|$ for the ideal fits $Q_i(\boldsymbol{y}_i - \boldsymbol{\beta}z_i)$ and the rates of $\|Q_i\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)z_i\|$.

Theorem 40.3
Suppose conditions I to IV hold and $\mathrm{var}(\boldsymbol{\epsilon}_i) \le (\sigma^*)^2 I_J$ for certain $0 < \sigma^* < \infty$. Then, for certain $\epsilon_{J,M}$ with $\lim_{M\to\infty}\lim_{J\to\infty}\epsilon_{J,M} \to 0$,

$$\max_{i\le n} P\left\{\|\hat{f}_i(\boldsymbol{x}_i) - f_i(\boldsymbol{x}_i)\|^2/J > M\left[\rho_{J,n}^* + (\sigma^*)^2 K_{J,n}^*/J\right]\right\} \le \epsilon_{J,M} .$$

In particular, if $K^*_{J,n} = O(1)J^{1/(2\alpha+1)}$ and $\rho^*_{J,n} = O(1)J^{2\gamma/(2\alpha+1)}$ for certain $0 < \gamma \le \alpha$, then $\|\hat{f}_i(\boldsymbol{x}_i) - f_i(\boldsymbol{x}_i)\|^2/J = O_P\left(J^{-2\gamma/(2\alpha+1)}\right)$, where the O_P is uniform in $i \le n$.

In the case of $\mathrm{var}(\boldsymbol{\epsilon}_i) = \sigma^2 I_J$, $\max_{i\le n} E\|Q_i(\boldsymbol{y}_i - \boldsymbol{\beta}z_i) - f_i(\boldsymbol{x}_i)\|^2/J \ge \max\left(\rho^*_{J,n}, \sigma^2 K^*_{J,n}/J\right)$ is the convergence rate for the ideal fits $Q_i(\boldsymbol{y}_i - \boldsymbol{\beta}z_i)$ for $f_i(\boldsymbol{x}_i)$. Theorem 3 asserts that $\hat{f}_i(\boldsymbol{x}_i)$ have the same convergence rates as the ideal fits. Thus, $\hat{f}_i(\boldsymbol{x}_i)$ achieve or nearly achieve the optimal rate of convergence for normalization under standard conditions. See the two examples in *Huang* et al. [40.21].

40.7 Concluding Remarks

The basic idea of TW-SLM normalization is to estimate the normalization curves and the relative gene effects simultaneously. The TW-SLM normalization does not assume that the normalization is constant as in the global normalization method, nor does it make the assumptions that the percentage of differentially expressed genes is small or that the up- and down-regulated genes are distributed symmetrically, as are required in the lowess normalization method [40.11]. This model puts normalization and significant analysis of gene expression in the framework of a high dimensional semiparametric regression model. We used a back-fitting algorithm to compute the semiparametric M-estimators in the TW-SLM. For identification of differentially expressed genes, we used an intensity-dependent variance model. This variance model is a compromise between the constant residual variance assumption used in the ANOVA method and the approach in which the variances of all the genes are treated as being different. We described two nonparametric methods for variance estimation. The first method is to smooth the scatter plot of the squared residuals versus the total intensity. The second one is to estimate the variance function jointly with the normalization curves and gene effects. For the example we considered in Sect. 40.6, the proposed method yields reasonable results when compared with the published results. Our simulation studies show that the TW-SLM normalization has better performance in terms of the mean squared errors than the lowess and spline normalization methods. Thus the proposed TW-SRM for microarray data is a powerful alternative to the existing normalization and analysis methods.

The TW-SLM is qualitatively different from the SRM. For microarray data, the number of genes J is always much greater than the number of arrays n. This fits the description of the well-known small-n large-p difficulty (we use p instead of J to be consistent with the phrase used in the literature). Furthermore, in the TW-SLM, both n and J play the dual role of sample size and number of parameters. That is, for estimating $\boldsymbol{\beta}$, J is the number of parameters, n is the sample size. But for estimating $\boldsymbol{f}$, n is the number of (infinite dimensional) parameters, J is the sample size for each f_i. On one hand, sufficiently large n is needed for the inference of $\boldsymbol{\beta}$. But a large n makes normalization more difficult, because then more nonparametric curves need to be estimated. On the other hand, sufficiently large J is needed for accurate normalization, but then estimation of $\boldsymbol{\beta}$ becomes more difficult. We are not aware of any other semiparametric models [40.33] in which both n and J play such dual roles of sample size and number of parameters. Indeed, here the difference between the sample size and the number of parameters is no longer as clear as in a conventional statistical model. This reflects a basic feature of microarray data in which self-calibration in the data is required when making statistical inference.

References

40.1 M. Schena, D. Shalon, R. W. Davis, P. O. Brown: Quantitative monitoring of gene expression patterns with a complementary cDNA microarray, Science **270**, 467–470 (1995)

40.2 P. O. Brown, D. Botstein: Exploring the new world of the genome with microarrays, Nat. Genet. **21**(1), 33–37 (1999)

40.3 P. Hedge, R. Qi, K. Abernathy, C. Gay, S. Dharap, R. Gaspard, J. Earle-Hughes, E. Snesrud, N. Lee, J. Quackenbush: A concise guide to cDNA microarray analysis, Biotechniques **29**, 548–562 (2000)

40.4 M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein: Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA **95**(25), 14863–14868 (1998)

40.5 T. R. Golub, D. K. Slonim, P: Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, E. S. Lander: Molecular classification of cancer:

Class discovery and class prediction by gene expression monitoring, Science **286**(5439), 531–537 (1999)

40.6 A. A. Alizadeh, M. B. Eisen, E. R. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, L. M. Staudt: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature **403**(6769), 503–511 (2000)

40.7 M. J. Callow, S. Dudoit, E. L. Gong, T. P. Speed, E. M. Rubin: Microarray expression profiling identifies genes with altered expression in HDL deficient mice, Gen. Res. **10**, 2022–2029 (2000)

40.8 Y. Chen, E. R. Dougherty, M. L. Bittner: Ratio-based decisions and the quantitative analysis of cDNA microarray images, J. Biomed. Opt. **2**, 364–374 (1997)

40.9 M. K. Kerr, M. Martin, G. A. Churchill: Analysis of variance for gene expression microarray data, J. Comp. Biol. **7**, 819–837 (2000)

40.10 T. B. Kepler, L. Crosby, K. T. Morgan: Normalization and analysis of DNA microarray data by self-consistency and local regression, Genome Biol. **3**(7), research0037.1–research0037.12 (2002)

40.11 Y. H. Yang, S. Dudoit, P. Luu, T. P. Speed: Normalization for cDNA microarray data. In: *Microarrays: Optical Technologies and Informatics*, Proceedings of SPIE, Vol. 4266, ed. by M. L. Bittner, Y. Chen, A. N. Dorsel, E. R. Dougherty (Int. Soc. Opt. Eng., San Diego 2001) pp. 141–152

40.12 G. C. Tseng, M.-K. Oh, L. Rohlin, J. C. Liao, W.-H. Wong: Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variation and assessment of gene effects, Nucleic Acids Res. **29**, 2549–2557 (2001)

40.13 D. B. Finkelstein, J. Gollub, R. Ewing, F. Sterky, S. Somerville, J. M. Cherry: Iterative linear regression by sector. In: *Methods of Microarray Data Analysis. Papers from CAMDA 2002*, ed. by S. M. Lin, K. F. Johnson (Kluwer Academic, Dordrecht 2001) pp. 57–68

40.14 J. Quackenbush: Microarray data normalization and transformation, Nat. Gen. (Suppl.) **32**, 496–501 (2002)

40.15 T. Park, S-G. Yi, S-H. Kang, S. Y. Lee, Y. S. Lee, R. Simon: Evaluation of normalization methods for microarray data, BMC Bioinformatics **4**, 33–45 (2003)

40.16 J. Fan, P. Tam, G. Vande Woude, Y. Ren: Normalization and analysis of cDNA micro-arrays using within-array replications applied to neuroblastoma cell response to a Cytokine, Proc. Natl. Acad. Sci. **101**, 1135–1140 (2004)

40.17 J. Fan, H. Peng, T. Huang: Semilinear high-dimensional model for normalization of microarray data: a theoretical analysis and partial consistency, J. Am. Stat. Assoc. **100**, 781–796 (2005)

40.18 W. S. Cleveland: Robust locally weighted regression and smoothing scatterplots, J. Am. Stat. Assoc. **74**, 829–836 (1979)

40.19 J. Huang, H.-C. Kuo, I. Koroleva, C.-H. Zhang, M. B. Soares: A Semi-linear Model for Normalization and Analysis of cDNA Microarray Data, Tech Report **321** (2003) Depart. of Stat., Univ. Iowa, Iowa City

40.20 J. Huang, C.-H. Zhang: Asymptotic analysis of a two-way semiparametric regression model for microarray data, Stat. Sin. **15**, 597–618 (2005)

40.21 J. Huang, D. L. Wang, C-H. Zhang: A two-way semilinear model for normalization and significant analysis of microarray data, J. Am. Stat. Assoc. **100**, 814–829 (2005)

40.22 G. Wahba: Partial spline models for semiparametric estimation of functions of several variables. In: *Statistical Analysis of Time Series*, Proceedings of the Japan U.S. Joint Seminar Tokyo (Inst. Stat. Mathematics, Tokyo 1984) pp. 319–329

40.23 R. F. Engle, C. W. J. Granger, J. Rice, A. Weiss: Semiparametric estimates of the relation between weather and electricity sales, J. Am. Stat. Assoc. **81**, 310–320 (1986)

40.24 P. Heckman: Spline smoothing in partly linear model, J. R. Stat. Soc. Ser. B **48**, 244–248 (1986)

40.25 H. Chen: Convergence rates for a parametric component in a partially linear model, Ann. Stat. **16**, 136–146 (1988)

40.26 P. Huber: *Robust Statistics* (Wiley, New York 1981)

40.27 L. Schumaker: *Spline Functions: Basic Theory* (Wiley, New York 1981)

40.28 T. Hastie, R. Tibshirani, J. Friedman: *The Elements of Statistical Learning* (Springer, New York 2001)

40.29 R Development Core Team: *R: A Language and Environment for Statistical Computing* (R Foundation Stat. Computing, Vienna 2003) http//www.R-project.org.

40.30 D. Ruppert, M. P. Wand, U. Holst, O. Hössjet: Local polynomial variance-function estimation, Technometrics **39**, 262–273 (1997)

40.31 J. Fan, Q. Yao: Efficient estimation of conditional variance functions in stochastic regression, Biometrika **85**, 645–660 (1998)

40.32 S. Dudoit, Y. H. Yang, M. J. Callow, T. P. Speed: Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments, Stat. Sin. **12**, 111–140 (2000)

40.33 P. J. Bickel, C. A. J. Klaassen, Y. Ritov, J. A. Wellner: *Efficient and Adaptive Estimation for Semiparametric Models* (Johns Hopkins Univ. Press, Baltimore 1993)

41. Latent Variable Models for Longitudinal Data with Flexible Measurement Schedule

This chapter provides a survey of the development of latent variable models that are suitable for analyzing unbalanced longitudinal data. This chapter begins with an introduction, in which the marginal modeling approach (without the use of latent variable) for correlated responses such as repeatedly measured longitudinal data is described. The concepts of random effects and latent variables are introduced at the beginning of Sect. 41.1. Section 41.1.1 describes the linear mixed models of Laird and Ware for continuous longitudinal response; Sect. 41.1.2 discusses generalized linear mixed models (with latent variables) for categorical response; and Sect. 41.1.3 covers models with multilevel latent variables. Section 41.2.1 presents an extended linear mixed model of Laird and Ware for multidimensional longitudinal responses of different types. Section 41.2.2 covers measurement error models for multiple longitudinal responses. Section 41.3 describes linear mixed models with latent class variables—the latent class mixed model that can be useful for either a single or multiple longitudinal responses. Section 41.4 studies the relationships between multiple longitudinal responses through structural equation models. Section 41.5 unifies all the above varieties of latent variable models under a single multilevel latent variable model formulation.

Longitudinal data consists of variables that are measured repeatedly over time. Longitudinal data can be collected either prospectively or retrospectively. The defining feature of longitudinal data is that the set of observations on one subject are likely to be correlated, and this within-subject correlation must be taken into account in order to make valid scientific inferences from the data. A frequently encountered problem in longitudinal studies is data that are missing due to missed visits or dropouts. As a result subjects often do not have a common set of visit times or they visit at nonscheduled times, thus longitudinal data may be highly unbalanced.

Except in the Introduction, the remaining sections are devoted to the study of the models along the lines of the Laird and Ware-style mixed model [41.1] and models with latent variables since they naturally handle unbalanced longitudinal data and of course these models are also useful for regularly spaced, balanced, repeatedly measured responses. Models suitable only for balanced longitudinal data as well as missing data models are not discussed in this chapter. All the models discussed in this chapter have been proved successful in practice. However, the models covered in this chapter only reflect the choice of illustration by the author and are by no means inclusive of all the variants and extensions of latent variable models.

Before moving onto the next section, let us first look at the marginal models that do not involve latent variables. Let Y_{ij} denote the longitudinal response for subject i $(i = 1, \ldots, N)$ at the j-th time point

$(j = 1, \ldots, n_i)$. For example, in the study of the evolution of the CD4+ lymphocyte in human immunodeficiency virus (HIV) positive subjects, longitudinal CD4+ cell counts can be modeled with the following *general linear model* for continuous response as:

$$Y_{ij} = \boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{\epsilon}_{ij} \,, \tag{41.1}$$

where $\boldsymbol{x}_{ij} = (x_{ij1}, \ldots, x_{ijp})^{\mathrm{T}}$ is a vector of fixed p covariates at j-th time point that includes the intercept of one, the linear term of time in months, the indicator of AZT (an anti-retrovirus drug) usage, the Karnovsky score, anemia, etc., $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-1})^{\mathrm{T}}$ are the coefficients for the intercept and the partial slopes, and ϵ_{ij} is an error term that has a zero mean and variance of σ^2. The correlation between ϵ_{ij} and $\epsilon_{ij'}$ is $\rho_{jj'}$ for $j \neq j'$. Let $\boldsymbol{Y}_i = (Y_{i1}, \ldots, Y_{in_i})^{\mathrm{T}}$ and $\mathbf{X}_i = (\boldsymbol{x}_{i1}^{\mathrm{T}}, \ldots, \boldsymbol{x}_{in_i}^{\mathrm{T}})^{\mathrm{T}}$, then the above model can be written in matrix notation as

$$\boldsymbol{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i \,.$$

So $\boldsymbol{Y}_i$ has mean vector $\mathbf{X}_i\boldsymbol{\beta}$ and variance matrix $\sigma^2\mathbf{R}_i$, where $\mathbf{R}_i$ is the correlation matrix for $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{i,n_i})^{\mathrm{T}}$.

For discrete responses, marginal models that extend the *generalized* linear models GLMs can be applied [41.2]. Marginal models model a *link* function of the population-average response, $E(Y_{ij})$, as a function of a common set of explanatory variables $\boldsymbol{x}$. The mean of the longitudinal response is modeled separately from the within-subject correlation that is usually assumed to be a function of the modeled marginal means and possibly additional parameters $\boldsymbol{\nu}$. The marginal model is specified as:

$$g(\mu_{ij}) = \boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\beta} \,, \tag{41.2}$$

where g is the monotone *link* function, $\mu_{ij} = E(Y_{ij})$. For example, g is an identity link for continuous Gaussian response, g can be a *logit* link for binary response. An attractive feature of the marginal model is that within-subject correlation does not have to be modeled explicitly, rather a class of generalized estimating equations (GEE) that gives consistent estimates of the $\boldsymbol{\beta}$ and their variance is used under some assumed working correlation matrices for within-subject dependence without specifying a multivariate distribution for $\boldsymbol{Y}_i$ [41.3].

The regression coefficients $\boldsymbol{\beta}$ in marginal models have population-average interpretation but any heterogeneity beyond the recorded covariates cannot be accounted for in marginal models. In models with latent variables, that are studied in following sections, heterogeneity among subjects in a subset of the regression coefficients, e.g., the intercept, are taken into account via subject-specific regression coefficients and/or covariates. In latent variable models, the covariate effects and within-subject association are modeled simultaneously. The concept of latent variables is a convenient way to represent statistical variation in terms of measurement error, random coefficients and variance components. Modeling the heterogeneity of a subset of regression coefficients not only reduces the extent of unexplained variation beyond the recorded explanatory variables but may be of interest in its own right. Estimates of the parameters in the latent variable models studied in this chapter can be obtained via the likelihood method with either the EM (expectation-maximization) algorithm or (adaptive) Gaussian quadrature.

41.1 Hierarchical Latent Variable Models for Longitudinal Data

One may consider longitudinal data as having a two-level structure, with repeated measurements (level 1) of a response variable being nested within subjects (level 2). Traditional fixed-effect analytical methods (e.g., analysis of variance) are limited in their treatment of the technical difficulties presented by nested designs, and in the questions they are able to address. Models that include random regression coefficients are more suited to the hierarchical data structure generally found in longitudinal data.

Latent variables are unobservable individual regression coefficients, predictors/covariates or response variables in regression models. Latent variables here are thus divided into the following three types, and sometimes a latent variable qualifies for more than one of the three types. The first type is called *random effects*, which model heterogeneity among subjects in a subset of the regression coefficients that vary from one subject to the next. A defining feature of random effects is that the individual regression coefficients are assumed to be a random sample from a common distribution so that a few parameters for the distribution characterize the behavior of the entire random coefficients. The second type is called *latent covariates*; these unobservable latent covariates have their own fixed regression coefficients that are called *factor loadings*. The third type is

called *latent responses*, which are further modeled on other fixed covariates and/or latent covariates.

Softwares for fitting latent variable models are abundant, although they may only handle a limited number of the models discussed in this chapter. Many researcher-written computer codes and softwares are also available for fitting complicated latent variable models. One omnipass software for fitting the latent variable models studied in this chapter is the STATA module generalized linear latent and mixed models (GLLAMM) developed over years by *Rabe-Hesketh* et al. [41.4]. For some models, it may take quite some computer and real time to fit. The computing time usually depends on the size of data, the number of structural levels in the data, and more critically on the number of latent variables involved. Software development will not be further discussed in this chapter.

41.1.1 Linear Mixed Model with a Single-Level Latent Variable

Since repeated measurements are obtained from each individual at different times, there may be considerable variation among individuals in the number and timing of observations. The resulting unbalanced data are typically not amenable to analysis using a general multivariate model such as (41.1) above, mainly due to the difficulty in specifying the covariance for $\boldsymbol{\epsilon}_i$ without the aid of latent variables. Although marginal models like (41.2) can handle unbalanced longitudinal data they do not model the heterogeneity across individual subjects beyond the recorded covariates. The model with random effects originally proposed by *Laird* and *Ware* [41.1] readily accommodates both the unbalanced nature of the data and the heterogeneity across subjects.

For continuous responses, their model is specified as:

$$\boldsymbol{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i \ . \tag{41.3}$$

Here, $\boldsymbol{Y}_i = (Y_{i1}, \cdots, Y_{in_i})^{\mathrm{T}}$, is the n_i-vector of longitudinal readings for subject i. Fixed covariates including the intercept and possibly deterministic functions of time (such as a linear term of time) from subject i are represented by the $n_i \times p$ matrix $\mathbf{X}_i = (\boldsymbol{x}_{i1}^{\mathrm{T}}, \cdots, \boldsymbol{x}_{in_i}^{\mathrm{T}})^{\mathrm{T}}$, with associated p-vector of coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-1})^{\mathrm{T}}$. The j-th row of $\mathbf{X}_i$, denoted $\boldsymbol{x}_{ij}^{\mathrm{T}}$, is thus a p-vector of covariate values measured at the j-th occasion. Covariates for random effects are denoted by the $n_i \times q$ matrix $\mathbf{Z}_i$, which is often a subset of $\mathbf{X}_i$ although does not have to be. The q-vector of individual regression coefficients, the random effects $\boldsymbol{b}_i = (b_{i1}, \ldots, b_{iq})^{\mathrm{T}}$, are taken to be independent, multi-normally distributed with mean $\mathbf{0}$ or non-zero (see the example of random intercept only that follows) and variance–covariance $\mathbf{D}$. The error $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{in_i})^{\mathrm{T}}$ is an n_i-vector that is uncorrelated with $\boldsymbol{b}_i$, independent normals with mean $\mathbf{0}$ and covariance matrix $\mathbf{R}_i$, which is often assumed to be a n_i-diagonal matrix of $\sigma^2\mathbf{I}_{n_i}$. Given the random effects, the timings of covariates and $\boldsymbol{Y}$ are assumed to be non-informative.

Marginally, the $\boldsymbol{Y}_i$ are independent normals with mean $\mathbf{X}_i\boldsymbol{\beta}$ and covariance matrix $\mathbf{R}_i + \mathbf{Z}_i\mathbf{D}\mathbf{Z}_i^{\mathrm{T}}$. For a single time point response Y_{ij}, the above model can be rewritten as:

$$Y_{ij} = \boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{z}_{ij}^{\mathrm{T}}\boldsymbol{b}_i + \epsilon_i \ . \tag{41.4}$$

It can be seen that the covariance between two responses measured at different times points j and j' within a subject is $z_{ij}\mathbf{D}\boldsymbol{z}_{ij'}^{\mathrm{T}} = \mathrm{cov}(Y_{ij}, Y_{ij'})$.

The covariates that have random effects in this model have the means of their effects absorbed into the fixed effects so that the mean of $\boldsymbol{b}_i$ can be conveniently assumed to be zero. For a linear growth model with random intercept only, the above model becomes

$$Y_{ij} = \beta_0 + \beta_1 t_{ij} + b_{i0} + \epsilon_{ij} \ ,$$

where t_{ij} is the time of j-th repeated measure for subject i, the random intercept b_{i0} is assumed to have independent normal distribution with mean zero and variance σ_b^2, and the error term ϵ_{ij} is assumed to be independent of b_{i0} and to be normally distributed with mean zero and variance σ^2. The same model can also be written as:

$$Y_{ij} = b_0^* + \beta_1 t_{ij} + \epsilon_{ij}$$

in which the random intercept b_0^* is assumed to have a non-zero mean of β_0 and variance σ_b^2. Notice that a random effect can either be represented as having a mean or as being deviations from the mean.

For this random intercept model, the within-subject correlation coefficient is $\sigma_b/\sqrt{\sigma^2 + \sigma_b^2}$. It should be pointed out that, for the linear mixed model (41.4), the population-average inference can be made readily from the fixed-effects part, that is, $E\boldsymbol{Y}_{\overline{X}} = \overline{\mathbf{X}}\boldsymbol{\beta}$, where $\overline{\mathbf{X}}$ is the population-average covariate values. Although the number of individual random regression coefficients is large, the additional parameters that need to be estimated beyond the fixed regression coefficients are only those involved in the variance of the individual regression coefficients, which are called *variance components*. The use of random effects not only allows individualized

growth trajectories, but also conveniently accommodates within-subject correlation. The model (41.4) can be fit with the readily available software such as the SAS proc MIXED, as well as many other commercial softwares through the (restricted) maximum-likelihood method.

41.1.2 Generalized Linear Model with Latent Variables

Harville and *Mee* [41.5] extended the aforementioned linear mixed model for continuous response to clustered ordinal data using a threshold probit model, which also turned out to be suitable for longitudinal ordinal response. Their model was motivated by a study in which cattle breeders were interested in comparing sire with respect to the difficulty experienced in the birth of their offspring. There are five ordinal difficulty levels for the response variable: no problem, slight difficulty, needed assistance, considerable force needed and extreme difficulty. Let Y_{ij} be the ordinal response for the j-th birth by sire i ($i = 1, \ldots, N$; $j = 1, \ldots, n_i$) that take a value from one of the ordered difficulty categories $1, \ldots, M$, where here $M = 5$. For the threshold probit model, it is often assumed that there is an unobserved latent variable η relating to the actual observed ordinal response Y. Here, a response occurs in category m ($Y_{ij} = m$) if the latent variable η_{ij} exceeds the threshold value θ_{m-1} but does not exceed the threshold value θ_m. The ordinal response is related to the latent variable via the following probit model:

$$P(Y_{ij} = m|\eta_{ij}) = \Phi\left(\frac{\theta_m - \eta_{ij}}{\sigma}\right) - \Phi\left(\frac{\theta_{m-1} - \eta_{ij}}{\sigma}\right) , \tag{41.5}$$

where Φ is the cumulative distribution of a standard normal and σ is the standard error of the residual ϵ_{ij} from the following linear mixed model for the latent response η:

$$\eta_{ij} = \boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{z}_{ij}^{\mathrm{T}}\boldsymbol{b}_i + \epsilon_{ij} .$$

This model is similar to the model (41.4) except Y_{ij} is now replaced by η_{ij}. It can be seen that η serves both as a latent covariate and a latent response.

The ordinal response can also be specified by a threshold cumulative probit model [41.6] as:

$$P(Y_{ij} \le m|\eta_{ij}) = \Phi\left(\frac{\theta_m - \eta_{ij}}{\sigma}\right) . \tag{41.6}$$

For binary, nominal or count data, Y_{ij}, can be modeled using the generalized linear mixed model (GLMM) [41.7], which is a direct generalization of the linear mixed model (41.4):

$$g(\mu_{ij}|\boldsymbol{b}_i) = \boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{z}_{ij}^{\mathrm{T}}\boldsymbol{b}_i , \tag{41.7}$$

where g is the *link* function of μ. g can be probit or logit binary Y_{ij}; g can be log for count Y_{ij}; g can be logit and $\mu_{ij} = \{P(Y_{ij} = m), m \in [1, \ldots, M]\}$ for nominal Y_{ij}. It should be noted that the marginal means of Y_{ij} are no longer $\boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\beta}$, but a more complicated function depending on the form of the link function g and on the mean of the response itself. Both SAS proc NLMIXED and the STATA add-on function GLLAMM [41.4] can fit the nonlinear mixed model with logit and probit links for binomial data, logit link for polychotomous data, probit and cumulative probit for ordinal data, and log link for Poisson data.

41.1.3 Model with Hierarchical Latent Variables

Consider, the example of the National Youth Survey data analyzed by *Duncan* et al. [41.8], where repeated measures were taken from individuals nested within households nested within geographical areas. The resulting data structure, therefore, consists of four levels: repeated observations within a subject (level 1), subjects (level 2), households (level 3), and geographical areas (level 4). The response variable of interest is recorded as a scale of substance use. A question that naturally arises is whether each level in the data structure has its own submodel representing the structural relations and variability occurring at that level.

Since there was no evidence of variation among the geographical areas, a three-level model without the fourth level is presented here. Let $j = 1, \ldots, n_i$, $i = 1, \ldots, N$ and $k = 1, \ldots, K$ index respectively the repeated observations within a subject, subjects within a household, and households. Let t_{kij} denote the j-th time point when the measurements for subject i in household k are taken. The level-1 within-individual growth model can be expressed as:

$$Y_{kij} = \boldsymbol{x}_{kij}^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{z}_{kij}^{\mathrm{T}}\boldsymbol{\eta}_{ki} + \boldsymbol{\epsilon}_{kij} , \tag{41.8}$$

where Y_{kij} is the response for subject i in household k at the j-th time point, $\boldsymbol{x}_{kij} = \boldsymbol{z}_{kij} = (1, t_{kij})^{\mathrm{T}}$ is a vector of covariates including the intercept and the time. The fixed coefficients vector $\boldsymbol{\beta} = (\beta_0, \beta_1)^{\mathrm{T}}$ includes the intercept and the slope of time for Y_{kij}. The errors $\boldsymbol{\epsilon}_{ki} = (\epsilon_{ki1}, \ldots, \epsilon_{kin_i})^{\mathrm{T}}$ are assumed to be normally distributed with mean zerp and covariance matrix $\mathbf{R}_i$,which is a diagonal matrix with diagonal elements σ^2.

The individual random effects vector $\boldsymbol{\eta}_{ki} = (\eta_{ki0}, \eta_{ki1})^{\mathrm{T}}$ including the intercept and the slope is modeled in the following level-2 model for individuals within the same household as:

$$\boldsymbol{\eta}_{ki} = \boldsymbol{b}_k + \boldsymbol{\xi}_{ki} \,, \tag{41.9}$$

where the entries in the household-level random effects vector $\boldsymbol{b}_k = (\boldsymbol{b}_{0k}, \boldsymbol{b}_{1k})^{\mathrm{T}}$ at level 3are assumed to be bivariate normal distributions with mean zero and covariance matrix $\mathbf{D}$, and where $\boldsymbol{\xi}_{ki}$ is the residual vector, which is independent across different subjects, uncorrelated with $\boldsymbol{b}_k$, and bivariate normal distributed with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$.

Models (41.8) and (41.9) can be combined into the following single model:

$$Y_{kij} = \boldsymbol{x}_{kij}^{\mathrm{T}}\boldsymbol{\beta} + z_{kij}^{\mathrm{T}}\boldsymbol{b}_k + z_{kij}^{\mathrm{T}}\boldsymbol{\xi}_{ki} + \boldsymbol{\epsilon}_{ki}, \tag{41.10}$$

in which, there are coefficients of fixed effects $\boldsymbol{\beta}$, random individual effects $\boldsymbol{\xi}_{ki}$ nested within a household and random household effects $\boldsymbol{b}_k$. The covariance between two repeated measures at different time points j and j' for a subject i is $z_{kij}\mathbf{D}z_{kij'}^{\mathrm{T}} + z_{kij}\boldsymbol{\Sigma}z_{kij'}^{\mathrm{T}}$ and the covariance between measurements from two different subjects i and i' within a household is $z_{kij}\mathbf{D}z_{kij'}^{\mathrm{T}}$. This example showed that the multilevel model can be formulated and estimated within the linear mixed model framework [41.9].

41.2 Latent Variable Models for Multidimensional Longitudinal Data

Longitudinal studies offer us an opportunity to develop detailed descriptions of the process of growth and development or of the course of progression of chronic diseases. Most longitudinal analyses focus on characterizing change over time in a single outcome variable and identifying predictors of growth or decline. Both growth and degenerative diseases, however, are complex processes with multiple markers of change, so that it may be important to model more than one outcome measure and to understand their relationship over time.

41.2.1 Extended Linear Mixed Model for Multivariate Longitudinal Responses

Lin et al. modeled multiple continuous longitudinal responses by using a mixed effects model for each of the longitudinal responses; the correlations among the different longitudinal responses were modeled through intercorrelated random effects across the mixed effects models; and the model naturally allows different measurement schedules for different types of longitudinal responses even within a same subject [41.10]. The model was illustrated with the data example from a trial of chemoprevention of cancer with β-carotene [41.11]. The trial was a randomized double-blind placebo-controlled trial with 264 patients whose primary objective was to determine whether supplemental β-carotene (50 mg/d) reduced recurrence of the primary tumors in patients cured from a recent early-stage head and neck cancer. The trial concluded that the β-carotene supplementation had no significant effect on second head and neck cancers. During the trial, blood samples of the patients were collected at about 0, 3 months, 12 months and yearly thereafter until 60 months. Several plasma nutrient concentrations were determined from the available blood samples. Analysis was focused on plasma concentrations of lycopene and lutein +zeaxanthin.

Let $i = 1, \ldots, N$ index the i-th subject, $k = 1, \ldots, K$ index the k-th type of the longitudinal response variables and $j = 1, \ldots, n_{ki}$ index the j-th time point when type k-th response is measured in subject i. For the above example $K = 2$ with $k = 1$ and 2 indexing lycopene and lutein+zeaxanthin, respectively. Let $\boldsymbol{x}_{kij}$ denote a p_k-vector of fixed effects covariates for the type k longitudinal response measured at the j-th time in subject i, which includes the intercept, the baseline plasma cholesterol concentration, the treatment assignment indicator of β-carotene, site (0 for Connecticut and 1 for Florida), age, sex, smoking status ($\{0, 0\}$ for nonsmoker, $\{1, 0\}$ for transient smoker and $\{0, 1\}$ for steady smoker), and linear and quadratic terms of time. The vector of random effects z_{kij} is a q_k-subvector of $\boldsymbol{x}_{kij}$ that includes the intercept and linear and quadratic terms of time. The model for the multiple longitudinal responses is specified as:

$$Y_{kij} = \boldsymbol{x}_{kij}^{\mathrm{T}}\boldsymbol{\beta}_k + z_{kij}^{\mathrm{T}}\boldsymbol{b}_{ki} + \epsilon_{kij} \,. \tag{41.11}$$

In the above specification, $\boldsymbol{x}_{kij}$ and z_{kij} are associated with fixed regression coefficients $\boldsymbol{\beta}_k$ and random coefficients $\boldsymbol{b}_{ki}$, respectively. The q_k-vector of random effects $\boldsymbol{b}_{ki}$ is assumed to be independent across the i and multinormally distributed with mean $\mathbf{0}$ and covariance $\mathbf{D}_{kk}$. The n_{ki}-vector of error term $\boldsymbol{\epsilon}_{ki} = (\epsilon_{ki1}, \ldots, \epsilon_{kin_{ki}})^{\mathrm{T}}$ is uncorrelated with $\boldsymbol{b}_{ki}$, independent normal with mean $\mathbf{0}$ and variance–covariance matrix $\mathbf{R}_{ki}$. The correlations

between different types of longitudinal responses within a same subject are built through the covariance of the random effects by assuming $\mathrm{cov}(\boldsymbol{b}_{ki}, \boldsymbol{b}_{k'i}) = \mathbf{D}_{kk'}$ for $k \neq k'$ and therefore $\mathrm{cov}(Y_{kij}, Y_{k'ij'}) = z_{kij}\mathbf{D}_{kk'}z_{k'ij'}^{\mathrm{T}}$. The covariance between the repeated measures of same type at two different time points j and j' is $z_{kij}\mathbf{D}_{kk}z_{kij'}^{\mathrm{T}}$

Let $\mathbf{X}_{Ki}$ and $\mathbf{Z}_{Ki}$ denote the covariate matrices for fixed and random effects, respectively.The model (41.11) can be re-expressed exactly as (41.3) if the covariates matrices and the variance–covariance matrices are rearranged in the following way:

$$\mathbf{X_i} = \begin{pmatrix} \mathbf{X}_{1i} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{X}_{Ki} \end{pmatrix}, \quad \mathbf{Z_i} = \begin{pmatrix} \mathbf{Z}_{1i} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{Z}_{Ki} \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{11} & \dots & \mathbf{D}_{1K} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{K1} & \dots & \mathbf{D}_{KK} \end{pmatrix}, \quad \mathbf{R_i} = \begin{pmatrix} \mathbf{R}_{1i} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{R}_{Ki} \end{pmatrix}.$$

Using these expressions, the model (41.11) becomes

$$\boldsymbol{Y_i} = \mathbf{X_i}\boldsymbol{\beta} + \mathbf{Z_i}\boldsymbol{b_i} + \boldsymbol{\epsilon_i}\,,$$

where $\boldsymbol{Y_i} = (\boldsymbol{Y}_{1i}^{\mathrm{T}}, \dots, \boldsymbol{Y}_{Ki}^{\mathrm{T}})^{\mathrm{T}}$ with $\boldsymbol{Y}_{ki} = (Y_{ki1}, \cdots, Y_{kin_{ki}})^{\mathrm{T}}$ being a long vector of all K longitudinal response for subject i, $\boldsymbol{\epsilon_i} = (\boldsymbol{\epsilon}_{1i}^{\mathrm{T}}, \dots, \boldsymbol{\epsilon}_{Ki}^{\mathrm{T}})^{\mathrm{T}}$ is a long vector of error terms, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{T}}, \dots, \boldsymbol{\beta}_K^{\mathrm{T}})^{\mathrm{T}}$ is a long vector of fixed coefficients and $\boldsymbol{b_i} = (\boldsymbol{b}_{1i}^{\mathrm{T}}, \dots, \boldsymbol{b}_{KiJ}^{\mathrm{T}})^{\mathrm{T}}$ is a long vector of random coefficients. This model has an identical expression and meaning as (41.3).

It is straightforward to see that by using proper link functions the model (41.11) can be further extended for mixed continuous and categorical longitudinal responses.

41.2.2 Measurement Error Model

Multiple outcomes are sometimes needed to jointly characterize an effect of interest properly. *Roy* considered the situation where multiple longitudinal outcomes are assumed to measure an underlying quantity of main interest from different perspectives [41.12]. Although separate linear mixed models can be fitted for each outcome, this approach is limited by the fact that it fails to borrow strength across the outcome variables. By exploiting the correlation structure with a multivariate longitudinal model, efficiency and power could be greatly increased. Since different outcomes are often measured on different scales and different units, it is of substantial interest to develop a statistical model to account for this special feature of the data. Correlation within each outcome over time and between outcomes on the same unit must be taken into account.

Roy analyzed the methadone treatment practices data in which methadone treatment is important in reducing illicit drug use and preventing HIV transmission and is effective when certain critical treatment practices are followed. The sampling unit is the treatment practice unit. The three longitudinal outcomes measuring the effectiveness of methadone treatment, including the maximum dose level [$\boldsymbol{Y}_1 = \log(\text{maximum dose})$], unit-average length of treatment ($\boldsymbol{Y}_2$), and percentage of clients receiving decreasing doses [$\boldsymbol{Y}_3 = \log(\text{percentage})$], were collected at three follow-up times. Analysis of this data set is challenging due to the fact that the outcome of major interest, the effective treatment practices level, is not observable, although several surrogates are available. In the following illustration, the same notation as for model (41.11) are used.

Suppose that the K longitudinal outcomes attempt to characterize a latent outcome of major interest, η_{ij}, e.g., the treatment practices level in unit i at the j-th follow-up time in the methadone example. One way to view this problem is that each type of observed outcome (Y_{kij}, $k = 1, \dots, K$) measures the latent variable η_{ij} with error. It is likely that the measurement error for each outcome from the same unit is correlated over time. A linear mixed model is assumed to relate Y_{kij} to η_{ij}:

$$Y_{kij} = \beta_{k0} + \beta_{k1}\eta_{ij} + b_{ki0} + \epsilon_{kij}\,, \tag{41.12}$$

where the measurement error term ϵ_{kij} is independent normal with mean zero and variance σ_k^2 and the type-specific random intercept b_{ki0} is independent between different units and assumed to have a normal distribution with mean zero and variance σ_{kb}^2. Correlation between the different types of the outcomes in an unit is due to the shared latent variable η_{ij}. The observed outcome Y_{kij} then measures the underlying true treatment practice evolution with error. The factor loading β_{k1} and the type-specific intercept β_{k0} are used to accommodate the fact that different types of outcomes have different scales. Each unit has its random intercept b_{ki0} for the type-k outcome, which is a random deviation from the type-specific intercept β_{k0}. For the sake of identifiability, β_{11} is set to one.

A linear mixed model is assumed to describe the effects of covariates on the latent variable η_{ij} of the underlying treatment practice:

$$\eta_{ij} = \boldsymbol{x}_{ij}^{\mathrm{T}}\boldsymbol{\alpha} + \boldsymbol{z}_{ij}^{\mathrm{T}}\boldsymbol{a}_i + \xi_{ij}\,, \tag{41.13}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{a}_i$ are defined similarly to $\boldsymbol{\beta}$ and $\boldsymbol{b}_i$ in the linear model (41.4), $\boldsymbol{a}_i$ is a $q-$vector with normal($\mathbf{0}$, $\mathbf{D}$) and the residual ξ_{ij} is distributed with independent normal$(0, \sigma^2)$. $\boldsymbol{x}_{ij}$ and $\boldsymbol{z}_{ij}$ are vectors of fixed and random covariates at the j-th time point for unit i that are defined similarly to those in the linear model (41.4) except that $\boldsymbol{x}_{ij}$ does do not contain the intercept.

It is often of substantial interest to estimate the unit-specific latent variables η_{ij}. The estimates of the latent effective practices score via the posterior mean $E(\boldsymbol{\eta}_i|\boldsymbol{Y}_i)$ can be used to identify the units whose treatment practices effectiveness are well below those of a typical unit. The model also provides a straightforward way to estimate and test for global covariate effects since the parameters $\boldsymbol{\alpha}$ represent the effects of the covariates on the overall effective treatment practices level in the methadone data. Estimates of the parameters in (41.12) and (41.13) can be obtained via the EM algorithm as described by Roy or using the STATA add-on function GLLAMM of *Rabe-Hesketh* et al. [41.4].

Extension of the model to allow mixed discrete and continuous outcomes is straightforward, e.g. the model (41.12), can be modified to allow discrete outcomes through the GLM formulation:

$$g_k(\mu_{kij}) = \beta_{k0} + \beta_{k1}\eta_{ij} + b_{ki0} \,,$$

where g_k is a link function specific to the type-k outcome and $\mu_{kij} = E(Y_{kij})$.

41.3 Latent Class Mixed Model for Longitudinal Data

The linear mixed model is a well-known method for incorporating heterogeneity (for example, subject-to-subject variation) into a statistical analysis for continuous responses. However heterogeneity cannot always be fully captured by the usual assumptions of normally distributed random effects. Latent class mixed models offer a way of incorporating additional heterogeneity which can be used to uncover distinct subpopulations, to incorporate correlated non-normally distributed outcomes and to classify individuals into *risk* classes. *Lin* et al. and *McCulloch* et al. used a latent class mixed model [41.13, 14] to model the trajectory of longitudinal prostate specific antigens (PSAs) before diagnosis of prostate cancer from a retrospective study of nutritional prevention of cancer (NPC) trials in which subjects were randomized to either selenium-supplement groups or the control group [41.15, 16]. Serial PSA levels were determined retrospectively from frozen blood samples that had been collected from all patients at successive clinic visits. The PSA data set that was analyzed consists of 1182 subjects with a highly variable number of readings (range 120, median 4) per subject at irregularly spaced intervals.

These latent class mixed models assume that there are K latent classes, with each class representing a subpopulation that has its own trajectories of longitudinal responses. Suppose we have N subjects indexed by $i = 1, \ldots, N$, and K latent classes labeled by $k = 1, \ldots, K$. We define $C_{ik} = 1$ if subject i is member of class k and 0 otherwise. The probability that subject i belongs to latent class k is described through the multinomial distribution of the class membership vector for subject i, $\boldsymbol{C}_i = (C_{i1}, \cdots, C_{iK})^{\mathrm{T}}$, modeled via a logit model with covariate vector $\boldsymbol{v}_i = (v_{i1}, \cdots, v_{im})^{\mathrm{T}}$ and associated class-specific coefficient vector $\boldsymbol{\phi}_k$:

$$\pi_{ik} = P(C_{ik} = 1) = \frac{\exp\left(\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{\phi}_k\right)}{\sum_{s=1}^{K}\exp\left(\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{\phi}_s\right)}, \qquad (41.14)$$

where π_{ik} denotes the probability that subject i belongs to latent class k and $\boldsymbol{\phi}_k$ is the coefficient vector for class k with $\boldsymbol{\phi}_1 = 0$.

Each subpopulation has its own model for the longitudinal response with subpopulation differences entering the mean:

$$\boldsymbol{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{b}_i + \mathbf{W}_i(\boldsymbol{\Gamma}\,\boldsymbol{C}_i) + \boldsymbol{\epsilon}_i \,. \qquad (41.15)$$

Here, $\boldsymbol{Y}_i, \mathbf{X}_i, \mathbf{Z}_i, \boldsymbol{\beta}, \boldsymbol{b}_i$ and $\boldsymbol{\epsilon}_i$ are defined in the same way as those in model (41.3). Covariates for class-specific effects are denoted by the $n_i \times p_w$ matrix $\mathbf{W}_i$, which has a similar structure to $\mathbf{X}_i$. There may be overlap of the covariate effects in $\mathbf{X}_i$, $\mathbf{Z}_i$ and $\mathbf{W}_i$. The class-specific regression parameters are in the $p_w \times K$ matrix $\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_K)$, with $\boldsymbol{\gamma}_k$ being a p_w-dimensional column vector containing the parameters specific to class k. Given $C_{ik} = 1$, we have $\boldsymbol{\Gamma}\,\boldsymbol{C}_i = \boldsymbol{\gamma}_k$ for $k = 2, \ldots, K$ and we take $\boldsymbol{\gamma}_1 = 0$ to assure identifiability.

The model (41.15) captures common characteristics of the longitudinal trajectories within a subpopulation through latent classes while accommodating the variability among subjects in the same class through random effects. The use of a mixture of multivariate normal distributions for the longitudinal response $\boldsymbol{Y}$ provides flexibility that allows non-normal distributions for random effects.

The variables included in the model are as follows. The covariate vector $\boldsymbol{v}$ used to predict class membership

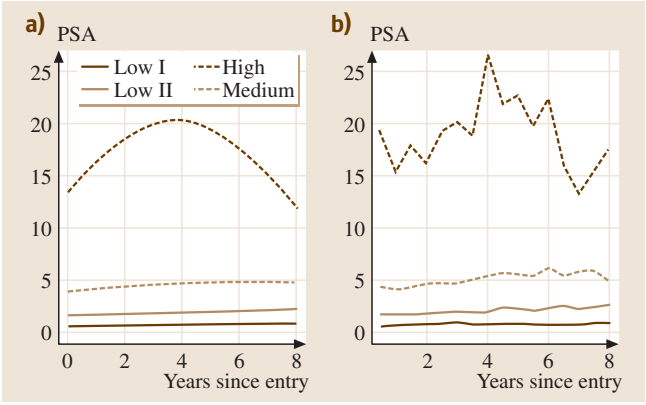


Fig. 41.1a,b Longitudinal trajectories of PSA for the four-class models. PSA values were fitted to the log-transformed data and then back-transformed to the original scale for plotting. The observed trajectories in the *right panel* are calculated by first dividing the time period into six-month intervals. For each subject, for each interval, the available PSA readings are averaged. The observed PSA trajectory for class k are calculated as averages weighted by each individual's estimated probability of class-k membership: **(a)** fitted trajectories for the four-class model; **(b)** observed trajectories for the four-class model

in (41.14) contains the treatment assignment indicator of selenium (Se) supplementation group, age at randomization, baseline PSA and Se level at randomization. The longitudinal biomarker value $\boldsymbol{Y}$ in (41.15) is the vector of log(PSA+1). The fixed effect covariate vector $\boldsymbol{X}$ contains the treatment assignment indicator, age at randomization and Se level at randomization, and linear and quadratic terms of visit time expressed in years since entry into the trial. The covariates for the random effects and class-specific effects, $\boldsymbol{Z}$ and $\boldsymbol{W}$, also contain an intercept and linear and quadratic terms of years since entry.

The four-class solution from the above latent class mixed model identifies fitted PSA trajectory classes that are labeled as "Low I","Low II", "Medium" and "High" Fig. 41.1. The majority classes "Low I" and "Low II" are characterized by a consistently low PSA level throughout the trial period. The "Medium" class has a higher PSA level than the two "Low" classes throughout the trial; the PSA level increases over time for this class. The minority class "High" has the highest PSA level at the beginning of the trial, and the predicted level of PSA increases over time until the fourth year after randomization and then decreases. In comparison the usual linear mixed model (41.4) would only be able to give one PSA trajectory that is rather flat.

Extension of the latent class mixed model to simultaneously modeling of multiple longitudinal responses is straightforward.

41.4 Structural Equation Model with Latent Variables for Longitudinal Data

Structural equation models (SEM) refer to the models that additionally specify the regression relationships among latent variables themselves. Models (41.9) and (41.13) can be regarded as SEMs.

Modeling growth within the SEM framework is a more recent approach for studying developmental trends. Because the SEM approach offers more flexibility in testing different research hypotheses about the developmental trend, many researchers have argued in favor of its superiority over some other analytic approaches [41.17, 18]. These models have provided researchers with an array of tools to interpret longitudinal data, understand developmental processes, and formulate new research questions.

Frosch et al. studied the relationship between tobacco and illicit drug use of cocaine and heroin among 166 methadone-maintained persons participating in a smoking-cessation intervention [41.19]. After completing a two-week screening period, participants were randomly assigned to one of four conditions: (a) contingency management (CM; a behavioral treatment in which participants receive increasingly valuable incentives for providing successive breath samples documenting smoking abstinence; $n = 44$); (b) relapse prevention (RP; a cognitive–behavioral group treatment providing educational and skills training information for smoking cessation; $n = 42$); (c) CM and RP combined ($n = 46$); and (d) a control condition in which participants received neither CM nor RP ($n = 43$). During the 12-week treatment period, participants provided urine and breath samples for heroin and cocaine toxicology and measurement of expired CO three times weekly (Monday, Wednesday, and Friday). The impact of use of heroin and cocaine on levels and changes in cigarette use was assessed with latent growth models in structural equations framework. The time axis is divided into two-week periods for the 12 weeks of treatment. Scales for the use of heroin, cocaine, and

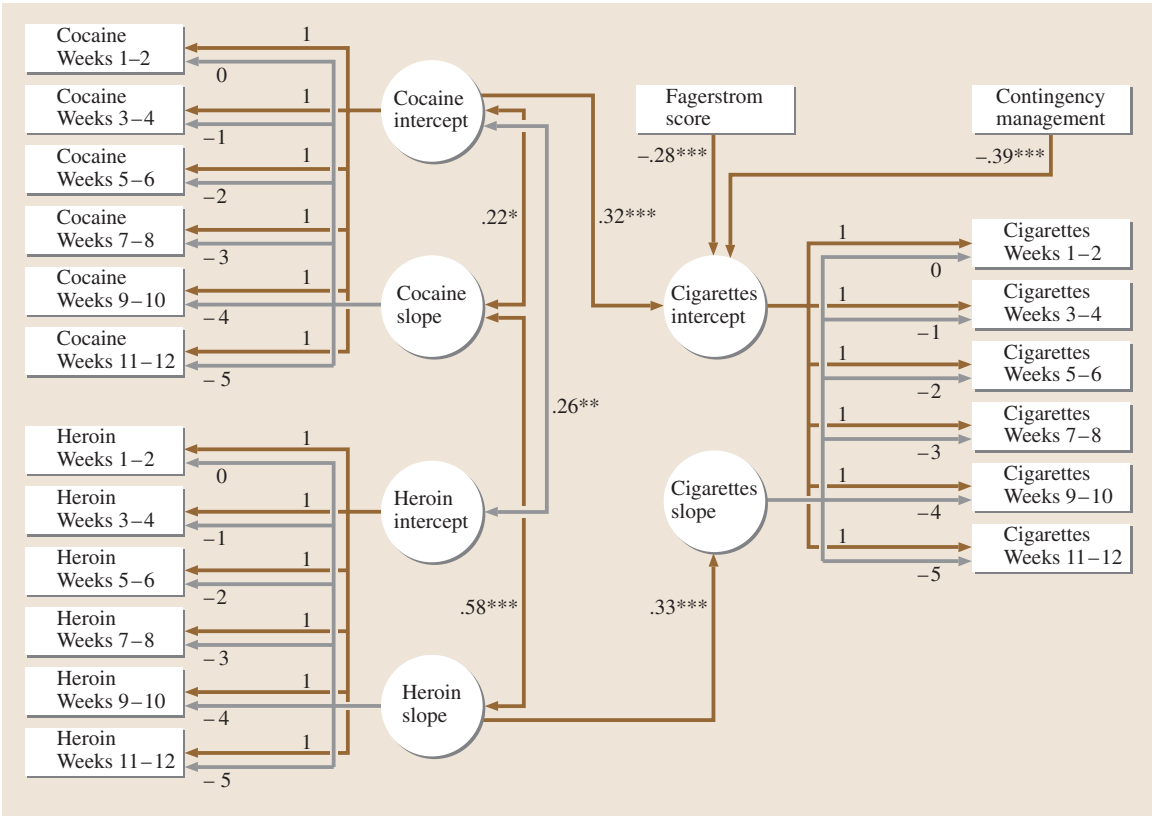


Fig. 41.2 Final latent-growth model presenting significant predictors of levels and trajectory of change in cigarette use over time. Intercepts were fixed to unity; slopes were hypothesized to be equal-interval linear trend coefficients. Double-headed arrows represent correlations; single-headed arrows represent regressions. Higher levels of cocaine use predicted higher levels of cigarette use; accelerated use of heroin predicted acceleration in use of cigarettes. *Circles* indicate latent variables; *rectangles* indicate measured variables. Parameter estimates are standardized. *Fagerstrom* = Fagerstrom test for nicotine dependence (after *Frosch* et al. [41.19])

cigarette were constructed for each of the two-week periods, and these scales were used as the longitudinal response variables.

Let $k = 1, 2, 3$ index the responses of cigarette, heroin and cocaine. For $k = 2$ or 3, an extended linear mixed model such as (41.11) is specified for heroin or cocaine response:

$$Y_{kij} = \boldsymbol{x}_{kij}^{\mathrm{T}}\boldsymbol{\beta}_k + \boldsymbol{z}_{kij}^{\mathrm{T}}\boldsymbol{\eta}_{ki} + \epsilon_{kij} \,,$$

where the fixed covariates $\boldsymbol{x}_{kij}$ include the intercept, the linear term in time in weeks and the treatment dummy variables, and where $\boldsymbol{z}_{kij}$ is a q_k-vector that includes only the intercept and the linear term in time; the model has exactly the same definitions as those of (41.11) except that the coefficients of the random effects $\boldsymbol{b}_{ki}$ in (41.11) are replaced by $\boldsymbol{\eta}_{ki}$ here. The model describes the possible improvement in heroin or cocaine use over the course of treatment and accounts for the repeated measures of the same type within the same subject with the random effects of intercept and slope included in $\boldsymbol{\eta}_{ki}$. The correlation between the responses of the two different types of heroin and cocaine use is modeled with the covariance of $\boldsymbol{\eta}_{2i}$ and $\boldsymbol{\eta}_{3i}$. The intercept represents the individual baseline level of use of cocaine or heroin. The slope represents the trend of the growth curve.

With the additional $p_{w,1}$-vector of fixed covariates of $\boldsymbol{w}_{1i}$ including the Fagerstrom score and contingency management measure, the slopes and intercepts of cocaine use and heroin use are used as the latent predictors in the following model to ascertain the impact of their initial levels and their own dynamic changes (slopes) on

predicting the slope and intercept of cigarette use:

$$\boldsymbol{\eta}_{1i} = \boldsymbol{\Lambda}_2\boldsymbol{\eta}_{2i} + \boldsymbol{\Lambda}_3\boldsymbol{\eta}_{3i} + \boldsymbol{\Gamma}_1\boldsymbol{w}_{1i} + \boldsymbol{\xi}_{1i}\,, \tag{41.16}$$

where $\boldsymbol{\Lambda}_k$ $(k = 2, 3)$ is a $q_k \times q_k$ ($q_k = 2$ here) diagonal matrix of factor loadings for $\boldsymbol{\eta}_{ki}$ with the s-th diagonal element being $\lambda_{k,s}$, $\boldsymbol{\Gamma}_1$ is a $q_1 \times p_{w,1}$ ($q_1 = p_{w,1} = 2$ here) matrix of regression coefficients associated with fixed covariates $\boldsymbol{w}_{1i}$ and ξ_{1i} is a q_1-vector of residuals.

The model (41.16) can be written as the following structural equation for the relationships among the latent variables:

$$\boldsymbol{\eta}_{\mathbf{i}} = \boldsymbol{\Lambda}\boldsymbol{\eta}_{\mathbf{i}} + \boldsymbol{\Gamma}\boldsymbol{w}_i + \boldsymbol{\xi}_i\,, \tag{41.17}$$

where $\boldsymbol{\eta}_{\mathbf{i}} = (\boldsymbol{\eta}_{1i}^{\mathrm{T}}, \ldots, \boldsymbol{\eta}_{Ki}^{\mathrm{T}})^{\mathrm{T}}$, $\boldsymbol{\Lambda}$ is a $\sum_{k=1}^{K} q_k \times \sum_{k=1}^{K} q_k$ upper-diagonal matrix of coefficients, $\boldsymbol{w}_i$ is a vector of $\sum_{k=1}^{K} p_{w,k}$ covariates, $\boldsymbol{\Gamma}$ is a $\sum_{k=1}^{K} q_k \times \sum_{k=1}^{K} p_{w,k}$ matrix of regression coefficients and $\boldsymbol{\xi}_i$ is a $\left(\sum_{k=1}^{K} q_k\right)$-vector of residuals. $\boldsymbol{\Lambda}$ is upper-diagonal, which implies that there are no simultaneous effects with latent variable 1 regressed on latent variable 2 and vice versa. The lower-level latent variables (e.g., the $\boldsymbol{\eta}_{1i}$ for cigarette use) come before the higher-level ones (e.g., the $\boldsymbol{\eta}_{2i}$ and $\boldsymbol{\eta}_{3i}$ for heroin and cocaine use) in the $\boldsymbol{\eta}$ vector, an the upper-diagonal $\boldsymbol{\Lambda}$ matrix ensures that lower-level latent variables can be regressed on higher ones but not the reverse since it would not make sense to regress a higher-level latent variable on a lower-level one. Some elements of the upper-diagonal matrix $\boldsymbol{\Lambda}$ can be additionally set to zero, which indicates that a latent variable does not depend on a corresponding higher-level one.

Using this structural equation model for the methadone-maintenance data, a significant relationship during the treatment period between rate of change in heroin and rate of change in tobacco use was revealed, with increased heroin use corresponding to increased tobacco use. Although levels of cocaine use were related to levels of tobacco use, there was no significant relationship between the rates of change of the two substances. *Frosch* et al.'s findings demonstrate the utility of latent growth models with the structural equation approach for analyzing short-term clinical trial data and strongly suggest that successful smoking cessation in this population requires a concurrent focus on reducing heroin use. The final model that *Frosch* et al. used is represented by a path diagram, as shown in Fig. 41.2.

41.5 Concluding Remark: A Unified Multilevel Latent Variable Model

In the case of hierarchical data including longitudinal data, the term *level* is often used to describe the position of a unit of observation within a hierarchy of units, typically reflecting the sampling design. Here level-1 units are nested in level-2 units, which are nested in level-3 units, a typical example being patients in clinics in regions. In this context, a random effect is said to vary at a given level, e.g. at the region level, if it varies between regions but, for a given region, is constant for all clinics and patients belonging to that region. If a repeated measure is taken on the patients and the regions are ignored, then time points are the level-1 units, patients are the level-2 units and clinics are the level-3 units. The multilevel models assume that lower-level units are conditionally independent given the higher-level latent variables and the explanatory variables. The latent variables at the same level are usually assumed to be mutually correlated whereas latent variables at different levels are independent. The aim of multilevel modeling is to analyze data simultaneously from different levels of the hierarchy. All the models discussed in this chapter can be regarded as special cases of a multilevel model with latent variables for longitudinal data.

The expression for the s-th element of $\boldsymbol{\eta}$ $(s = 1, \ldots, \sum_{k=1}^{K} q_k)$ in an SEM such as (41.16) can be substituted into the expression for $(s-1)$-th element, which can be substituted into the expression for $(s-2)$-th element, and so forth. (i. e., recursive substitution). Then, using similar notational definitions to those documented by *Rabe–Hesketh* et al. [41.4], we obtain an equation of the following form for longitudinal data with constraints among the parameters:

$$g(\mu_{kij}) = \boldsymbol{x}_{kij}^{\mathrm{T}}\boldsymbol{\beta} + \sum_{l=2}^{L}\sum_{s=1}^{q_l} \boldsymbol{z}_{kij,s}^{(l),\mathrm{T}}\boldsymbol{\lambda}_s^{(l)}\eta_{ki,s}^{(l)}\,, \tag{41.18}$$

where $l = 1, \ldots, L$ indexes the L levels, there are q_l latent variables at level l, $\eta_{ki,s}^{(l)}$ is the s-th latent variable for subject i in the type-k response at level l, $\boldsymbol{x}_{kij}$ and $\boldsymbol{z}_{kij,s}^{(l)}$ are two vectors of explanatory variables associated with fixed and latent variables and $\boldsymbol{\lambda}_s^{(l)}$ is the vector of factor loadings for the s-th latent variable $\eta_{ki,s}^{(l)}$ in level l. In the general form of equation (41.18), the latent

variables η can be continuous or discrete (e.g., latent classes).

Multilevel modeling techniques offer researchers the opportunity not only to analyze their data in a more technically appropriate manner than traditional single-level methods allow, but also to extend the range of research questions to a level with more contextual richness and complexity.

References

41.1 N. M. Laird, J. H. Ware: Random-effects models for longitudinal data, Biometrics **38**, 963–974 (1982)

41.2 P. J. Diggle, P. Heagerty, K.-Y. Liang, S. L. Zeger: *Analysis of Longitudinal Data*, 2nd edn. (Oxford Univ. Press, Oxford 2002)

41.3 K.-Y. Y. Liang, S. L. Zeger: Longitudinal data analysis using generalized linear models, Biometrika **73**, 13–22 (1986)

41.4 S. Rabe-Hesketh, A. Skrondal, A. Pickles: *GLLAMM Manual*, U.C. Berkeley Division of Biostatistics Working Paper Series, Vol. 160 (2004) http://www.gllamm.org

41.5 D. A. Harville, R. W. Mee: A mixed-model procedure for analyzing ordered categorical data, Biometrics **40**, 393–408 (1984)

41.6 J. Catalano P. Bivariate modelling of clustered continuous and ordered categorical outcomes, Stat. Med. **16**, 883–900 (1997)

41.7 C. E. McCulloch, S. R. Searle: *Generalized, Linear, and Mixed Models* (Wiley, New York 2001)

41.8 T. E. Duncan, S. C. Duncan, H. Okut, L. A. Strycker, F. Li: An extension of the general latent variable growth modeling framework to four levels of the hierarchy, Struct. Equ. Model. **9**(3), 303–326 (2002)

41.9 A. Skrondal, S. Rabe-Hesketh: *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models* (Chapman Hall/CRC, New York 2004)

41.10 H. Q. Lin, C. E. McCulloch, S. T. Mayne: Maximum likelihood estimation in the joint analysis of time-to-event and multiple longitudinal variables, Stat. Med. **21**, 2369–2382 (2002)

41.11 S. T. Mayne, B. Cartmel, M. Baum, G. Shor-Posner, B. G. Fallon, K. Briskin, J. Bean, T. Z. Zheng, D. Cooper, C. Friedman, W. J. Goodwin: Randomized trial of supplemental beta-carotene to prevent second head and neck cancer, Cancer Res. **61**, 1457–1463 (2001)

41.12 J. Roy: Latent variable models for longitudinal data with multiple continuous outcomes, Biometrics **56**, 1047–1054 (2000)

41.13 H. Q. Lin, B. W. Turnbull, C. E. McCulloch, E. H. Slate: Latent class models for joint analysis of longitudinal biomarker and event process data: Application to longitudinal prostate-specific antigen readings and prostate cancer, J. Am. Stat. Assoc. **97**, 53–65 (2002)

41.14 C. E. McCulloch, H. Lin, E. H. Slate, B. W. Turnbull: Discovering subpopulation structure with latent class mixed models, Stat. Med. **21**, 417–429 (2002)

41.15 L. C. Clark, G. F. Combs Jr., B. W. Turnbull, E. H. Slate, D. K. Chalker, J. Chow, L. S. Davis, R. A. Glover, G. F. Graham, E. G. Gross, A. Krongrad, J. L. Lesher, H. K. Park, B. B. Sanders, C. L. Smith, J. R. Taylor: Effects of selenium supplementation for cancer prevention in patients with carcinoma of the skin, J. Am. Med. Assoc. **276**, 1957–1963 (1996)

41.16 L. C. Clark, B. Dalkin, A. Krongrad, G. F. Combs Jr., B. W. Turnbull, E. H. Slate, R. Witherington, J. H. Herlong, E. Janosko, D. Carpenter, C. Borosso, S. Falk, J. Rounder: Decreased incidence of prostate cancer with selenium supplementation: results of a double-blind cancer prevention trial, Brit. J. Urol. **81**, 730–734 (1998)

41.17 J. J. McArdle: A latent difference score approach to longitudinal dynamic analysis. In: *Structural Equation Modeling: Present and Future*, ed. by R. Cudeck, S. DuToit, D. Sörbom (Scientific Software International, Lincolnwood, IL 2001) pp. 341–380

41.18 J. J. McArdle, E. Ferrer-Caja, F. Hamagami, R. W. Woodcock: Comparative longitudinal structural analyses of the growth and decline of multiple intellectual abilities over the life span, Devel. Psychol. **38**, 115–142 (2002)

41.19 D. L. Frosch, J. A. Stein, S. Shoptaw: Use latent-variable models to analyze smoking cessation clinical trial data: an example among the methadone maintained, Exp. Clin. Psychopharmacol. **10**, 258–267 (2002)

42. Genetic Algorithms and Their Applications

The first part of this chapter describes the foundation of genetic algorithms. It includes hybrid genetic algorithms, adaptive genetic algorithms and fuzzy logic controllers. After a short introduction to genetic algorithms, the second part describes combinatorial optimization problems including the knapsack problem, the minimum spanning tree problem, the set-covering problem, the bin-packing problem and the traveling-salesman problem; these are combinatorial optimization studies problems which are characterized by a finite number of feasible solutions. The third part describes network design problems. Network design and routing are important issues in the building and expansion of computer networks. In this part, the shortest-path problem, maximum-flow problem, minimum-cost-flow problem, centralized network design and multistage process-planning problem are introduced. These problems are typical network problems and have been studied for a long time. The fourth section describes scheduling problems. Many scheduling problems from manufacturing industries are quite complex in nature and very difficult to solve by conventional optimization techniques. In this part the flow-shop sequencing problem, job-shop scheduling, the resource-constrained projected scheduling problem and multiprocessor scheduling are introduced. The fifth part introduces the reliability design problem, including simple genetic algorithms for reliability optimization, reliability design with redundant units and alternatives, network reliability design and tree-based network topology design. The sixth part describes logistic problems including the linear transportation problem, the multiobjective transportation problem, the bicriteria transportation problem with fuzzy coefficients and supply-chain management network design. Finally, the last part describes location and allocation problems including the location–allocation problem, capacitated plant-location problem and the obstacle location–allocation problem.

42.1 Foundations of Genetic Algorithms

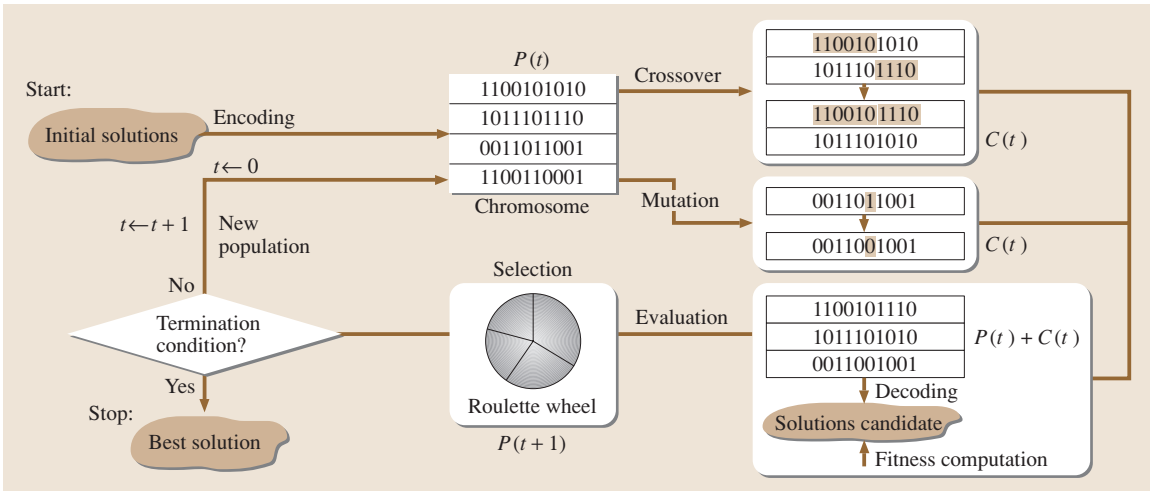


Fig. 42.1 The general structure of genetic algorithms

Recently, genetic algorithms have received considerable attention regarding their potential as an optimization technique for complex problems and have been successfully applied in the area of industrial engineering. The well-known applications include scheduling and sequencing, reliability design, vehicle routing location, transportation and many others.

42.1.1 General Structure of Genetic Algorithms

Genetic algorithms are stochastic search algorithms based on the mechanism of natural selection and natural genetics. Genetic algorithms, in contrast to conventional search techniques, start with an initial set of random solutions called the population. Each individual in the population is called a chromosome, encoding a solution to the problem at hand. A chromosome is a string of symbols, usually but not necessarily, a binary bit string. The chromosomes *evolve* through successive iterations, called generations. During each generation, the chromosomes are *evaluated*, using some measures of fitness [42.1]. To create the next generation, new chromosomes, called offspring, are formed by either merging two chromosomes from the current generation using a *crossover* operator or modifying a chromosome using a mutation operator. A new generation is formed by selecting, according to the fitness values, some of the parents and offspring, and rejecting others so as to keep the population size constant.

Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which we hope represents the optimum or suboptimal solution to the problem

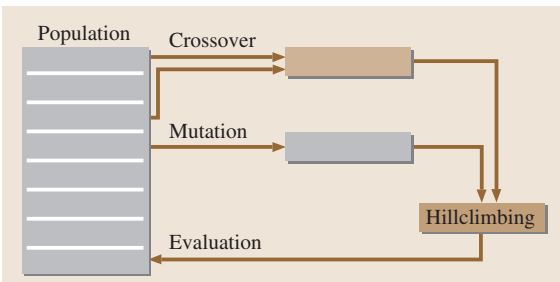


Fig. 42.2 General structure of hybrid genetic algorithms

when decoded. Let $P(t)$ and $C(t)$ be parents and offspring in the current generation t; the general structure of genetic algorithms Fig. 42.1 is described as follows:

```
procedure: genetic algorithms
begin
  t ← 0;              // t: generation number
  initialize P(t);    // P(t): population of individuals
  evaluate P(t);
  while (not termination condition) do
        crossover P(t) to yield C(t) ; // C(t): offspring
        mutation P(t) to yield C(t);
        evaluate C(t) ;
        select P(t+1) from P(t) and C(t) ;
        t ← t+1;
  end
end
```

Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combining both chromosomes' features. A simple way to achieve crossover would be to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point.

Mutation is a background operator, which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes.

42.1.2 Hybrid Genetic Algorithms

Genetic algorithms (GAs) have proved to be a versatile and effective approach for solving optimization problems. Nevertheless, there are many situations in which the simple GA does not perform particularly well, and various methods of have been proposed [42.2].

One of the most common forms of hybrid genetic algorithms is to incorporate local optimization as an add-on extra to the canonical GA loop of recombination and selection [42.3, 4]. With the hybrid approach, local optimization such as hill-climbing is applied to each newly generated offspring to move it to a local optimum before injecting it into the population. Genetic algorithms are used to perform global exploration among the population while heuristic methods are used to perform local exploitation around chromosomes. Because of the complementary properties of genetic algorithms and conventional heuristics, the hybrid approach often outperforms either method operating alone. Some work has been done to reveal the natural mechanism behind such a hybrid approach, among which is Lamarckian evolution. Let $P(t)$ and $C(t)$ be parents and offspring in the current generation t. The general structure of hybrid genetic algorithms is described as follows; see Fig. 42.2.

```
procedure: hybrid genetic algorithms
begin
  t ← 0 ;             // t: generation number
  initialize P(t);    // P(t): population of individuals
  evaluate P(t);
  while (not termination condition) do
        crossover P(t) to yield C(t);
                    // C(t): offspring
        mutation P(t) to yield C(t);
        locally search C(t);
        evaluate C(t);
        selection P(t+1) from P(t) and C(t);
        t  ←  t+1 ;
  end
end
```

42.1.3 Adaptive Genetic Algorithms

There are two basic approaches to applying the genetic algorithms to a given problem: 1) to adapt a problem to the genetic algorithms, 2) to adapt the genetic algorithms to a problem.

Genetic algorithms were first created as a kind of generic and weak method featuring binary encoding and binary genetic operators. This approach requires a modification of the original problem into an appropriate from suitable for the genetic algorithms, as shown in Fig. 42.3.

To overcome such problems, various nonstandard implementations of the genetic algorithm have been created for particular problems, which leave the problem unchanged and adapt the genetic algorithms by modifying a chromosome representation of a potential solution and applying appropriate genetic operators, as shown in Fig. 42.4. This approach has been successfully applied in the area of industrial engineering and is becoming the main approach in recent applications of genetic algorithms [42.5].

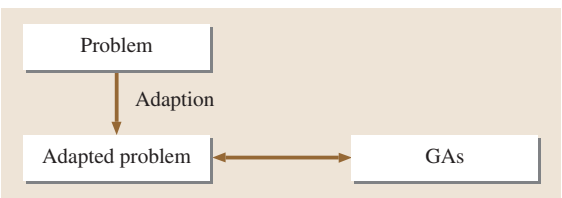


Fig. 42.3 Adapt a problem to the genetic algorithms

42.1.4 Fuzzy Logic Controller

Fuzzy logic is much closer in spirit to human thinking and natural language than the traditional logical systems. In essence, the fuzzy logic controller provides an algorithm which can convert a linguistic control strategy based on expert knowledge into an automatic control strategy. In particular, this methodology appears very useful when the processes are too complex for analysis by conventional techniques or when the available sources of information are interpreted qualitatively, inexactly, or with uncertainty [42.3].

The pioneering work to extend the fuzzy logic technique to adjust the strategy parameters of genetic algorithms dynamically was carried out by *Xu* and *Vukovich* [42.5]. The main idea is to use a fuzzy logic controller to compute new strategy parameter values that will be used by the genetic algorithms. A fuzzy logic controller is comprised of four principal components:

1. a knowledge base,
2. a fuzzification interface,
3. an inference system,
4. a defuzzification interface.

The experts' knowledge is stored in the knowledge base in the form of linguistic control rules. The inference system is the kernel of the controller, which provides an approximate reasoning based on the knowledge base. The generic structure of a fuzzy logic controller is shown in Fig. 42.5.

42.1.5 Multiobjective Optimization Problems

During the last two decades, genetic algorithms have received considerable attention regarding their potential as a novel approach to multiobjective optimization problems, known as evolutionary multiobjective optimization or genetic multiobjective optimization.

Multiobjective optimization problem with q objectives and m constraints will be formulated as follows:

$$\max \left[z_1 = f_1(\boldsymbol{x}), z_2 = f_2(\boldsymbol{x}), \ldots, z_q = f_q(\boldsymbol{x})\right] , \tag{42.1}$$

$$\text{s.t.} \quad g_i(\boldsymbol{x}) \le 0, \quad i = 1, 2, \ldots, m . \tag{42.2}$$

A. The Concept of a Pareto Solution

In most existing methods, Pareto solutions are identified at each generation and are only used to calculate fitness values or ranks for each chromosome. No mechanism is provided to guarantee that the Pareto solutions generated during the evolutionary process enter the next generation. A special pool for preserving the Pareto solutions is added onto the basic structure of genetic algorithms. At each generation, the set of Pareto solutions $E(t)$ is updated by deleting all dominated solutions and adding all newly generated Pareto solutions [42.5]. The overall structure of the approach is given as follows:

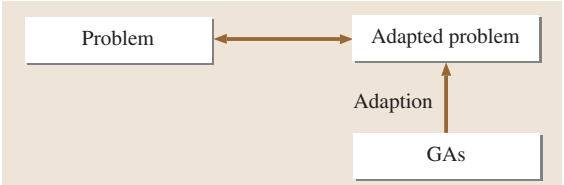


Fig. 42.4 Adapt the genetic algorithms to a problem

procedure: Pareto genetic algorithms
begin
  $t \leftarrow 0$; // t: generation number
  initialize $P(t)$; // $P(t)$: population of individuals
  objective $P(t)$;
  create Pareto $E(t)$;
  fitness $eval(P)$;
  while (not termination condition) **do**
    crossover $P(t)$ to yield $C(t)$;
      //$P(t)$: population of individuals
    mutation $P(t)$ to yield $C(t)$;
    objective $C(t)$;
    update Pareto $E(P, C)$;
    fitness $eval(P, C)$;
    selection $P(t+1)$ from $P(t)$ and $C(t)$;
    $t \leftarrow t+1$;
  end
end

B. Adaptive Weight Approach

Gen and *Cheng* proposed an adaptive weights approach which utilizes some useful information from the current population to readjust weights to obtain a search pressure towards a positive ideal point [42.6, 7].

For the examined solutions at each generation, we define two extreme points: the maximum extreme point z^+ and the minimum extreme pint z^- in criteria space as follows:

$$z^+ = \left(z_1^{\max}, z_2^{\max}, \ldots, z_q^{\max}\right) , \tag{42.3}$$

$$z^- = \left(z_1^{\min}, z_2^{\min}, \ldots, z_q^{\min}\right) , \tag{42.4}$$

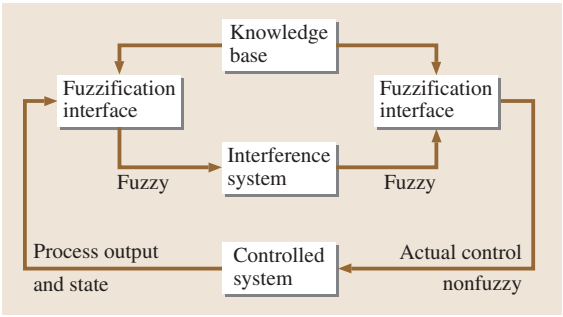


Fig. 42.5 Generic structure of a fuzzy logic controller

where $z_k^{\min}$ and $z_k^{\max}$ are the maximal value and minimal value for objective k in the current population. Let P denote the set of the current population. For a given individual $\boldsymbol{x}$, the maximal value and minimal value for each objective are defined as

$$z_k^{\max} = \max\{f_k(\boldsymbol{x}) | \boldsymbol{x} \in P\}, \quad k = 1, 2, \ldots, q\,, \tag{42.5}$$

$$z_k^{\min} = \min x\{f_k(\boldsymbol{x}) | \boldsymbol{x} \in P\}, \quad k = 1, 2, \ldots, q\,. \tag{42.6}$$

The hyperparallelogram defined by the two extreme points is a minimal hyperparallelogram containing all current solutions. The two extreme points are renewed at each generation. The maximum extreme point will gradually approximate the positive ideal point. The adaptive weight for objective k is calculated by

$$w_k = \frac{1}{z_k^{\max} - z_k^{\min}}, \quad k = 1, 2, \ldots, q\,. \tag{42.7}$$

For a given individual $\boldsymbol{x}$, the weighted-sum objective function is given by

$$z(\boldsymbol{x}) = \sum_{k=1}^{q} w_k \left(z_k - z_k^{\min}\right) \tag{42.8}$$

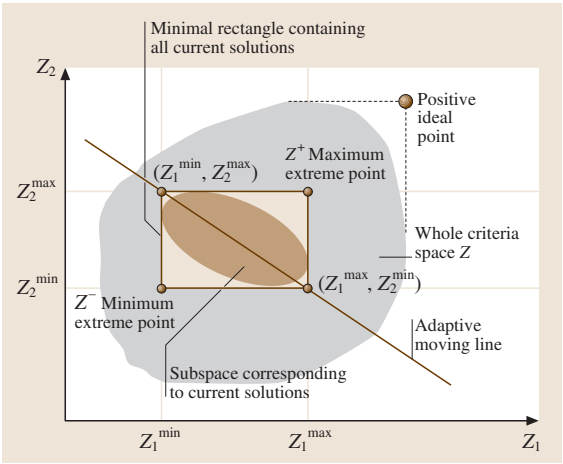


Fig. 42.6 Adaptive weights and adaptive hyperplane

$$= \sum_{k=1}^{q} \frac{z_k - z_k^{\min}}{z_k^{\max} - z_k^{\min}} \tag{42.9}$$

$$= \sum_{k=1}^{q} \frac{f_k(\boldsymbol{x}) - z_k^{\min}}{z_k^{\max} - z_k^{\min}}\,. \tag{42.10}$$

As the extreme points are renewed at each generation, the weights are renewed accordingly. Figure 42.6 is a hyperplane defined by the following extreme points in the current solutions

$$(z_1^{\max}, z_2^{\min}, \ldots, z_k^{\min}, \ldots, z_q^{\min})\,, \tag{42.11}$$

$$\ldots$$

$$(z_1^{\min}, z_2^{\min}, \ldots, z_k^{\max}, \ldots, z_q^{\min})\,, \tag{42.12}$$

$$\ldots$$

$$(z_1^{\min}, z_2^{\min}, \ldots, z_k^{\min}, \ldots, z_q^{\max})\,. \tag{42.13}$$

It is an adaptive moving line defined by the extreme points $(z_1^{\max}, z_2^{\min})$ and $(z_1^{\min}, z_2^{\max})$, as shown Fig. 42.6. The rectangle defined by the extreme points $(z_1^{\max}, z_2^{\min})$ and $(z_1^{\min}, z_2^{\max})$ is the minimal rectangle containing all current solutions.

42.2 Combinatorial Optimization Problems

Combinatorial optimization studies problems which are characterized by a finite number of feasible solutions. An important and widespread area of application concerns the efficient use of scarce resources to increase productivity. Typical problems include set covering, bin packing, knapsack, quadratic assignment, minimum spanning tree, machine scheduling, sequencing and balancing, cellular manufacturing design, vehicle routing, facility location and layout, traveling-salesman problem, and so on.

42.2.1 Knapsack Problem

Suppose that we want to fill up a knapsack by selecting some objects among various objects (generally called *items*). There are n different items available and each item j has a *weight* of w_j and a *profit* of p_j. The knapsack can hold a weight of at most W. The problem is to find an optimal subset of items so as to maximize the total profit subject to the knapsack's weight capacity. The profits, weights and capacity are positive integers [42.8].

Let x_j be binary variables given by

$$x_j = \begin{cases} 1 & \text{if item } j \text{ is selected}, \\ 0 & \text{otherwise}. \end{cases} \tag{42.14}$$

The knapsack problem can be mathematically formulated as

$$\max \sum_{j=1}^{n} p_j x_j \,, \tag{42.15}$$

$$\text{s.t.} \sum_{j=1}^{n} w_j x_j \le W \,, \tag{42.16}$$

$$x_j = 0 \quad \text{or } 1 \; j = 1, 2, \ldots, n \,. \tag{42.17}$$

Binary Representation Approach

The binary string is a natural representation for the knapsack problem, where one means the inclusion and zero the exclusion of one of the n items from the knapsack. For example, a solution for the 10-item problem can be represented as the following bit string:

$$x = (x_1 x_2 \cdots x_{10}) \\ (0\,1\,0\,1\,0\,0\,0\,0\,1\,0) \,,$$

meaning that items 2, 4 and 9 are selected for inclusion in the knapsack.

42.2.2 Minimum Spanning Tree Problem

Consider a connected undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ is a finite set of *vertices* representing terminals or telecommunication stations etc., $E = \{e_{ij} | e_{ij} = (v_i, v_j), v_i, v_j \in V\}$ is a finite set of *edges* representing connections between these terminals or stations. Each edge has an associated positive real number denoted by $W = \{w_{ij} | w_{ij} = w(v_i, v_j), w_{ij} > 0, v_i, v_j \in V\}$ representing distance, cost and so on. The vertices and edges are sometimes referred to as *nodes* and *links* respectively [42.9].

Based on their different backgrounds, many researchers have proposed varieties of spanning tree problems with some constraints on them, such as the spanning tree problem with a degree constraint, the stochastic spanning tree problem, the quadratic spanning tree problem, the multi-criteria spanning tree problem and the spanning tree problem with a constraint on the number of leaves or leaf-constrained spanning tree problem [42.10, 11].

A spanning tree is a minimal set of edges from E that connects all the vertices in V and therefore at least one spanning tree can be found in graph G. The minimum spanning tree is just one of the spanning trees whose total weight of all edges is minimal. It can be formulated as

$$\min z(x) = \sum_{i=1}^{n-1} \sum_{j=2}^{n} w_{ij} x_{ij} \,, \tag{42.18}$$

$$\text{s.t.} \sum_{i=1}^{n-1} \sum_{j=2}^{n} x_{ij} = n - 1 \,; \tag{42.19}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j > 1}} x_{ij} \le |S| - 1, \; S \subseteq V \backslash \{1\}, \; |S| \ge 2 \,, \tag{42.20}$$

$$x_{ij} = 0 \text{ or } 1 \,, \; i = 1, 2, \ldots, n-1, \\ = 2, 3, \ldots, n \,, \tag{42.21}$$

where

$$x_{ij} = \begin{cases} 1, & \text{if edge } (i, j) \text{ is selected in a spanning tree} \\ 0, & \text{otherwise} \end{cases} \tag{42.22}$$

and T is a set of the spanning trees of graph G.

A. Tree Encodings

For the minimum spanning tree (MST) problem, the method of encoding a tree is critical for the genetic algorithm approach because the solution should be a tree.

If we associate an index k with each edge, i. e., $E = \{e_k\}$, $k = 1, 2, \ldots, K$, where K is the number of edges in a graph, a bit string can represent a candidate solution by indicating which edges are used in a spanning tree, as illustrated in Fig. 42.7.

B. Genetic Approach

Representation. The chromosome representation for a spanning tree should contain, implicitly or explicitly, the degree on each vertex. Among the several tree encodings, only the Prüfer number encoding explicitly

contains the information of vertex degree, i.e. that any vertex with degree d will appear exactly $d-1$ times in the encoding. Thus the Prüfer number encoding is adopted.

Crossover and Mutation. Prüfer number encoding can still represent a tree after any crossover or mutation operations. Simply, the one-point crossover operator is used, as illustrated in Fig. 42.8. Mutation is performed as random perturbation within the permissive integer from 1 to n (n is the number of vertices in graph). An example is given in Fig. 42.9

42.2.3 Set-Covering Problem

The problem is to cover the rows of an m-row/n-column zero–one matrix by a subset of columns at minimal cost. Considering a vector $\boldsymbol{n}$ that x_j is $0-1$ variable that takes on the 3 value 1, if item j is selected (with a cost $c_j > 0$). The set-covering the problem is then formulated as

$$\min z(x) = \sum_{j=1}^{n} c_j x_j \,, \tag{42.23}$$

$$s.t. \sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1, 2, \ldots, m \,, \tag{42.24}$$

$$x_j \in \{0, 1\}, \quad j = 1, 2, \ldots, n \,. \tag{42.25}$$

Genetic Approach

Representation. The fitness of an individual $f(x)$ is calculated simply by

$$f(x) = \sum_{j=1}^{n} c_j x_j \,. \tag{42.26}$$

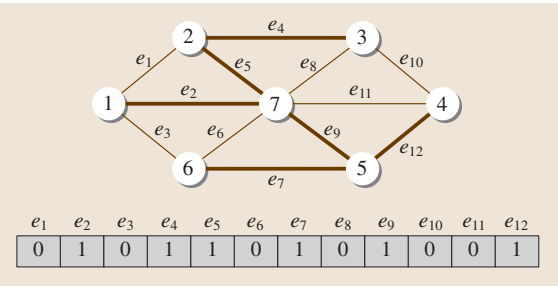


Fig. 42.7 A graph with its edge encoding for a spanning tree

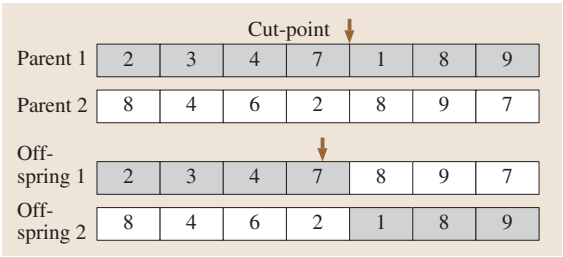


Fig. 42.8 Illustration of the crossover operation

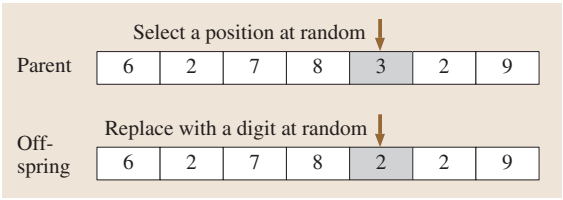


Fig. 42.9 Illustration of the mutation operation

The initial population can be generated randomly.

Genetic Operators. *Beasley* and *Chu* proposed a generalized fitness-based crossover operator called the fusion operator [42.9].

Let P_1 and P_2 be the parent strings. Let f_{P_1} and f_{P_2} be the fitness values of the parent strings P_1 and P_2, respectively. Let C be the child string. The fusion operator works as follows:

Procedure: Fusion Operator.

Step 1. $i = 1$.
Step 2. If $P_1[i] = P_2[i]$, then $C[i] \leftarrow P_1[i] = P_2[i]$.
Step 3. If $P_1[i] \neq P_2[i]$, then
(a) $C[i] \leftarrow P_1[i]$
with probability $p = f_{P_2}/(f_{P_1} + f_{P_2})$.
(b) $C[i] \leftarrow P_2[i]$ with probability $1 - p$.
Step 4. If $i = n$, stop; otherwise, set $i \leftarrow i + 1$ and go to step 1.

42.2.4 Bin-Packing Problem

The bin-packing problem consists of placing n objects into a number of bins (at most n bins). Each object has a weight ($w_i > 0$) and each bin has a limited bin capacity ($c_i > 0$). The problem is to find the best assignment of objects to bins such that the total weight of the objects in each bin does not exceed its capacity and the number of bins used is minimized.

A mathematical formulation for the bin-packing problem is given as follows [42.8]:

$$\min z(\boldsymbol{y}) = \sum_{i=1}^{n} y_i \,, \tag{42.27}$$

$$\text{s.t.} \sum_{j=1}^{n} w_j x_{ij} \le c_i y_i \,, \quad i \in N = \{1, 2, \ldots, n\} \,, \tag{42.28}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \,, \quad j \in N \,, \tag{42.29}$$

$$y_i = 0 \text{ or } 1, \quad i \in N \,, \tag{42.30}$$

$$x_{ij} = 0 \text{ or } 1, \quad i,\ j \in N \,, \tag{42.31}$$

where

$$y_i = \begin{cases} 1, & \text{if bin } i \text{ is used} \\ 0, & \text{otherwise} \,, \end{cases} \tag{42.32}$$

$$x_{ij} = \begin{cases} 1, & \text{if object } j \text{ is assigned to bin } i \\ 0, & \text{otherwise} \,. \end{cases} \tag{42.33}$$

Genetic Approach

Representation. The most straightforward approach is to encode the membership of objects in the solution. For instance, the chromosome 1 4 2 3 5 2 would encode a solution where the first object is in bin 1, the second in bin 4, the third in bin 2, the fourth in bin 3, the fifth in bin 5 and the sixth in bin 2. This representation for the bin-packing problem is illustrated in Fig. 42.10.

Genetic Operators

Procedure: Crossover [42.12].

Step 1. Select at random two crossing sites, delimiting the crossing section, in each of the two parents.

Step 2. Inject the contents of the crossing section of the first parent at the first crossing site of the second parent.

Step 3. Eliminate all objects now occurring twice from the bins they were members of in the second parent, so that the *old* membership of these objects gives way to the membership specified by the *new* injected bins. Consequently, some of the *old* groups coming from the second parent are altered.

Step 4. If necessary, adapt the resulting bins, according to the hard constraints and the cost function to optimize.

Step 5. Apply steps 2–4 to the two parents with their roles permuted to generate the second child.

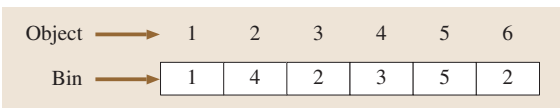


Fig. 42.10 Representation of membership of objects

42.2.5 Traveling-Salesman Problem

The traveling-salesman problem (TSP) is one of the most widely studied combinatorial optimization problems. Its statement is deceptively simple: a salesman seeks the shortest tour through n cities.

For example, a tour of a nine-city TSP

$$3-2-5-4-7-1-6-9-8$$

is simply represented as follows:

$$[3-2-5-4-7-1-6-9-8] \,.$$

This representation is also called a path representation or order representation. This representation may lead to illegal tours if the traditional one-point crossover operator is used, therefore many crossover operators have been investigated for it. Another method is the random keys representation. This representation encodes a solution with random numbers from (0,1). These values are used as sort keys to decode the solution.

For example, a chromosome for a nine-city problem may be

$$[0.23 \;\; 0.82 \;\; 0.45 \;\; 0.74 \;\; 0.87 \;\; 0.11 \;\; 0.56 \;\; 0.69 \;\; 0.78]$$

Where position i in the list represents city i. The random number in position i determines the visiting order of city i in a TSP tour. We sort the random keys in ascending order to get the following tour:

$$6-1-3-7-8-4-9-2-5$$

Genetic Approach

Representation. Permutation representation is perhaps the most natural representation of a TSP tour, where cities are listed in the order in which they are visited [42.13, 14].

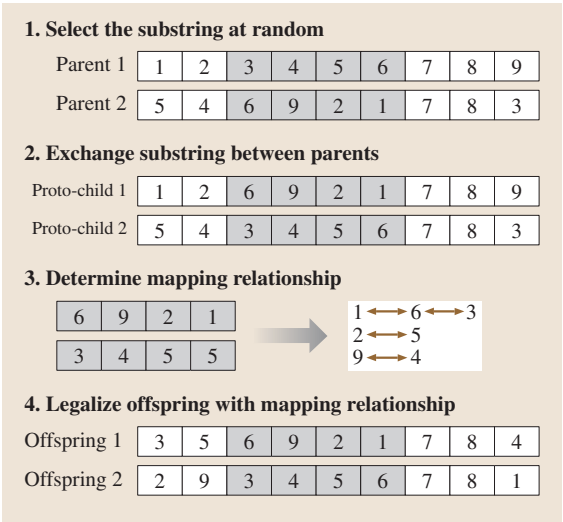


Fig. 42.11 Illustration of the PMX operator

Crossover Operators

Procedure: partial-mapped crossover (PMX) [42.14].

Step 1. Select two positions along the string uniformly at random.
Step 2. Exchange two substrings between parents to produce proto-children.
Step 3. Determine the mapping relationship between two mapping sections.

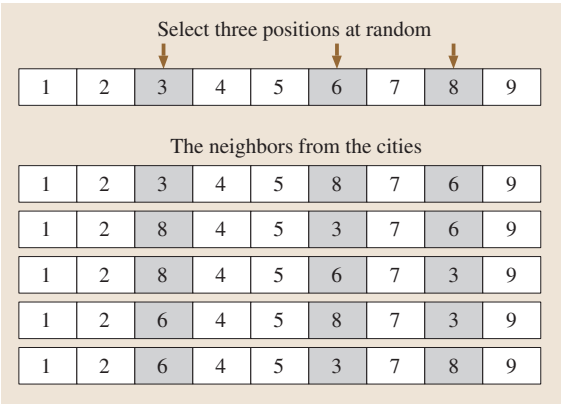


Fig. 42.12 Illustration of the heuristic mutation operator

Step 4. Legalize offspring with the *mapping relationship*.

The procedure is illustrated in Fig. 42.11.

Mutation Operators

Procedure: heuristic mutation [42.15, 16].

Step 1. Pick n genes at random.
Step 2. Generate neighbors according to all possible permutation of the selected genes.
Step 3. Evaluate all neighbors and select the best one as offspring.

The procedure is illustrated in Fig. 42.12.

42.3 Network Design Problems

Network design and routing are one of important issues in the building and expansion of computer networks. Many ideas and methods have been proposed and tested in the past two decades. Recently, there is an increasing interest in applying genetic algorithms to problems related to computer network [42.17].

42.3.1 Shortest-Path Problem

An undirected graph $G = (V, E)$ comprises a set of nodes $V = \{1, 2, \cdots, n\}$ and a set of edges $E \in V \times V$ connecting nodes in V. Corresponding to each edge, there are two nonnegative numbers c_{ij}^1 and c_{ij}^2 representing the cost and distance, or other items of interest, from node i to node j. A path from node i to node j is a sequence of edges (i, l), (l, m), $\cdots$, (k, j) from E in which no node appears more than once. A path can also be equivalently represented as a sequence of nodes $(i, l, m, \cdots, k, j)$. For the example given in Fig. 42.13, $(1, 4)$, $(4, 3)$, $(3, 5)$, $(5, 6)$ is a path from node 1 to node 6. The node representation is $(1, 4, 3, 5, 6)$.

Let 1 denote the initial node and n denote the end node of the path. Let x_{ij} be an indicator variable defined

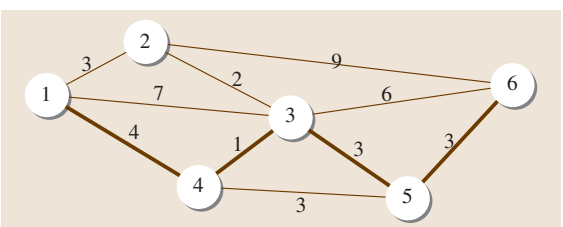


Fig. 42.13 Simple undirected graph with six nodes and 10 edges

as follows:

$$x_{ij} = \begin{cases} 1\,, & \text{if edge } (i,\,j) \text{ is included in the path} \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{42.34}$$

The bicriteria shortest-path problem can be formulated as follows:

$$\min z^1(x) = \sum_i \sum_j c_{ij}^1 x_{ij}\,, \tag{42.35}$$

$$\min z^2(x) = \sum_i \sum_j c_{ij}^2 x_{ij}\,, \tag{42.36}$$

$$\text{s.t.} \sum_j x_{ij} \le 2\,, \quad \forall i \in V\,, \tag{42.37}$$

$$\sum_{j \ne k} x_{ij} \ge x_{ik}\,, \quad \forall (i,k) \in E, \quad \forall i \in V \backslash \{1, n\}\,, \tag{42.38}$$

$$\sum_j x_{1j} = \sum_j x_{jn} = 1\,, \quad \forall i,\ j \in V\,, \tag{42.39}$$

$$x_{ij} = x_{ji}\,, \quad \forall (i,j) \in E\,, \tag{42.40}$$

$$0 \le x_{ij} \le 1\,, \quad \forall (i,j) \in E\,. \tag{42.41}$$

Genetic Approach

Priority-Based Encoding [42.18–20]. The position of a gene is used to represent a node and the value is used to represent the priority of the node for constructing a path among the candidates. The encoding method is denoted by priority-based encoding. The path corresponding to a given chromosome is generated by a sequential node-appending procedure, beginning from the specified node 1 and terminating at the specified node n.

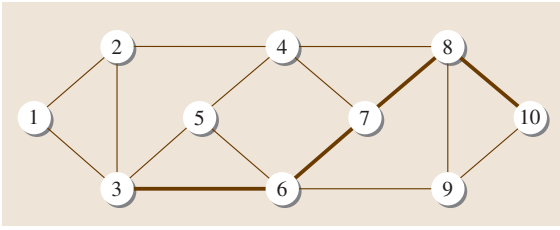


Fig. 42.14 Simple undirected graph with 10 nodes and 16 edges

Position: node ID	1	2	3	4	5	6	7	8	9	10
Value: priority	7	3	4	6	2	5	8	10	1	9

Fig. 42.15 Example of priority-based encoding

Consider the undirected graph shown in Fig. 42.14. Suppose we are going to find a path from node 1 to node 10. An encoding of the instance is given in Fig. 42.15. At the beginning, we try to find a node for the position next to node 1. Nodes 2 and 3 are eligible for the position, which can be easily fixed according to the adjacency relation among nodes. The priorities of them are 3 and 4, respectively. Node 3 has the highest priority and is put into the path. The possible nodes next to node 3 are nodes 2, 5 and 6. Because node 6 has the largest priority value, it is put into the path. Then we form the set of nodes available for the next position and select the one with the highest priority among them. These steps are repeated until we obtain a complete path (1, 3, 6, 7, 8, 10).

For an n-node problem, let Ω be a set containing integers from 1 up to n, that is, $\Omega = \{1, 2, \ldots, n\}$, let p_i denote the priority for node i, which is a random integer exclusively from the set Ω. Priorities p_i of all nodes satisfy the following conditions:

$$p_i \ne p_j, \quad p_i,\ p_j \in \Omega, \quad i \ne j,\ i, j = 1, 2, \ldots, n \tag{42.42}$$

Then the priority-based encoding can be formally defined as

$$[p_1 \ \ p_2 \ \ \cdots \ \ p_n]\,.$$

Genetic Operators

Here the position-based crossover operator proposed by *Syswerda* is adopted [42.21]. It can be viewed as a kind of uniform crossover operator for integer permutation representation together with a pairing procedure, as shown in Fig. 42.16. Essentially, it takes some genes from one parent at random and fills the vacuum position with genes from the other parent using a left-to-right scan. The swap mutation operator is used here, which simply selects two positions at random and swaps their contents as shown in Fig. 42.17.

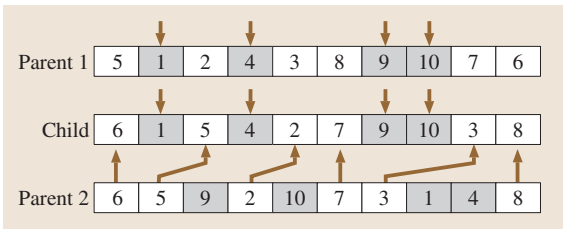


Fig. 42.16 Position-based crossover operator

42.3.2 Maximum-Flow Problem

There have been many applications of this problem in the real world. One of them is to determine the maximum flow through a pipeline network. Assume that oil should be shipped from the refinery (the source) to a storage facility (the sink) along arcs of the network. Each arc has a capacity which limits the amount of flow along that arc. Here, we want to determine the largest possible flow that can be sent from the refinery to the storage facility with the restriction that no arc (pipe) capacity can be exceeded. MXF has also been applied to some other applications such as: the problem of selecting sites for an electronic message-transmission system and dynamic flows in material-handling systems [42.22–24].

A mathematical formulation for the bin-packing problem is given by:

$$\max f \,, \tag{42.43}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} f, & \text{if} \quad i = 1 \\ 0, & \text{if} \quad i = 2, 3, \\ & \quad \ldots, m-1 \\ -f, & \text{if} \quad i = m \end{cases} , \tag{42.44}$$

$$0 \le x_{ij} \le u_{ij}, \quad i,\ j = 1, 2, \ldots, m \,, \tag{42.45}$$

where f is the amount of flow in the network from node 1 to node m and u_{ij} is arc capacities.

Genetic Approach

The priority-based encoding method is used to represent the chromosome. The chromosome here is represented by m-digit numbers that are generated randomly. Each number represents the priority of the node.

Crossover. As the first step in the crossover operation, we generate random numbers γ_k in the range [0, 1] ($k = 1, 2, \ldots, popSize$). Next, we select the chromosomes v_k to which the crossover operation will be applied. If $\gamma_k < p_C$ then the crossover operation will be applied to chromosome v_k.

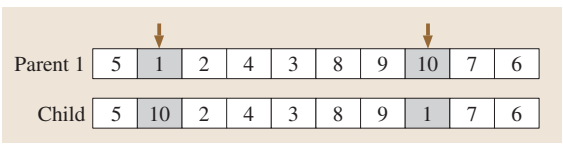


Fig. 42.17 Swap mutation operator

Mutation. Similarly, the first step in the mutation operation is to generate a random γ_r in the range [0, 1], ($r = 1, 2, \ldots, popSize$).

If $\gamma_r < p_M$ then the chromosome $v_k \quad (l = (r/m + 1))$ is chosen for the mutation operation.

42.3.3 Minimum-Cost-Flow Problem

The minimum-cost-flow problem (MCF) is known as a useful type of network optimization problem. It consists of finding the minimum-cost flows in the networks. For this problem, we are given a directed network $G = (X,\ A)$ in which each arc connecting node i and j in the network is associated with a cost c_{ij} and a capacity u_{ij}. A feasible solution to the MCF problem should satisfy two constraints. First, the flow through each arc should satisfy the capacity constraint. Second, the conservation of flow in all nodes should also be preserved. The conservation of flows here means that the flow into a node must equal the flow out of the node. The common objective is to determine the feasible network flow that minimizes the total cost.

A mathematical formulation for the bin-packing problem is given by:

$$\min z = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \,, \tag{42.46}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = b_i, i = 1, \ldots, m \tag{42.47}$$

$$x_{ij} \ge 0, i, j = 1, \ldots, m \,, \tag{42.48}$$

where x_{ij} is the flow through an arc and c_{ij} is the unit shipping cost along the arc. Equation (42.46) is called the flow-conservation or Kirchhoff equation and indicates that flow may be neither created nor destroyed in the network.

Genetic Approach

Representation. The chromosome here is represented by m-digit numbers generated randomly. Each number represents the priority of the node respectively.

Crossover. The crossover is done by selecting two chromosome randomly. We use the partially matched crossover (PMX) method for the crossover operation.

Mutation. Mutation here is done by selecting a chromosome at random. Two bit positions of the chromosome are exchanged.

42.3.4 Centralized Network Design

Consider a complete, undirected graph $G = (V,\ E)$, let $V = \{1, 2, \cdots, n\}$ be the set of nodes representing terminals. Denote the central site or *root* node as node 1, and let $E = \{(i, j)|i, j \in V\}$ be the set of edges representing all possible telecommunication wiring. For a subset of nodes $S(\subseteq V)$, define $E(S) = \{(i, j)|i, j \in S\}$ as the edges whose endpoints are both in S. Define the following binary decision variables for all edges $(i, j) \in E$:

$$x_{ij} = \begin{cases} 1\,, & \text{if edge } (i, j) \text{ is selected} \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{42.49}$$

Let c_{ij} be the fixed cost with respect to edge (i, j) in the solution, and suppose that d_i represents the demand at each node $i \in V$, where by convention the demand of the root node is $d_1 = 0$. Let $d(S)$, $S \subseteq V$ denote the sum of the demands of nodes of S. The subtree capacity is denoted with κ. The centralized network design problem can be formulated as follows [42.10]:

$$\min z = \sum_{i=1}^{n-1}\sum_{j=2}^{n} c_{ij}x_{ij}\,, \tag{42.50}$$

$$s.t. \sum_{i=1}^{n-1}\sum_{j=2}^{n} x_{ij} = 2(n-1)\,, \tag{42.51}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le 2[|S| - \lambda(S)]\,,$$

$$S \subseteq V\backslash\{1\}\,, |S| \ge 2\,, \tag{42.52}$$

$$\sum_{i\in U}\sum_{j\in U} x_{ij} \le 2(|U| - 1)\,,\quad U \subset V\,,$$

$$|U| \ge 2\,,\quad \{1\} \in U\,, \tag{42.53}$$

$$x_{ij} = 0 \text{ or } 1\,,\quad i = 1, 2, \ldots, n-1,$$

$$j = 2, 3, \ldots, n\,. \tag{42.54}$$

Equality (42.51) is true of all spanning trees: a tree with n nodes must have $n-1$ edges. Inequality (42.53) is a standard inequality for spanning trees: if more than $|U|-1$ edges connect the nodes of a subset U, then the set U must contain a cycle. The parameter $\lambda(S)$ refers to the *bin-packing number* of set S, namely, the number of bins of size κ needed to pack the nodes of items of size d_i for all $i \in S$. These constraints are similar to those for inequality (42.53), except that they reflect the capacity constraint: if the set S does not contain the root node, then the nodes of S must be contained in at least $\lambda(S)$ different subtrees of the root.

Up to now, all heuristic algorithms for this problem are only focused on how to deal with the constraints to make the problem simpler to solve. In the cutting plane algorithms or branch-bound algorithm, the network topology of the problem are usually neglected. As a result, it leads in an exponential explosion of constraints.

In Fig. 42.18, node ID is the node number based on the depth-first search (DFS) and the degree at node ID is the number of connecting nodes.

Genetic Approach

To solve the centralized network design problem by using a genetic algorithm, a tree-based permutation encoding method is adopted to encode the candidate solutions, as illustrated in Fig. 42.18.

42.3.5 Multistage Process Planning

The multistage process planning (MPP) system usually consists of a series of machining operations, such as turning, drilling, grinding, finishing, and so on, to transform a part into its final shape or product. The whole process can be divided into several stages. At each stage, there are a set of similar manufacturing operations. The MPP problem is to find the optimal process planning among all possible alternatives given certain criteria such as minimum cost, minimum time, maximum quality, or under several of these criteria.

For an n-stage MPP problem, let s_k be some state at stage k, $D_k(s_k)$ be the set of possible states to be chosen at stage k, $k = 1, 2, \ldots, n$, x_k be the decision variable to determine which state to choose at stage k; obviously $x_k \in D_k(s_k)$, $k = 1, 2, \ldots, n$. Then the MPP problem can be formulated as follows:

$$\min_{\substack{x_k\in D_k(s_k)\\ k=1,2,\ldots,n}} V(x_1,\ x_2, \cdots,\ x_n) = \sum_{k=1}^{n} v_k(s_k,\ x_k)\,, \tag{42.55}$$

where $v_k(s_k,\ x_k)$ represents the criterion to determine x_k under state s_k at stage k, usually defined as a real number such as cost, time, or distance.

Genetic Approach

Representation. The MPP solution can be concisely encoded in a state permutation format by concatenating all the set states of stages. This state permutation encoding has a one-to-one mapping for the MPP problem. The probability of randomly producing a process

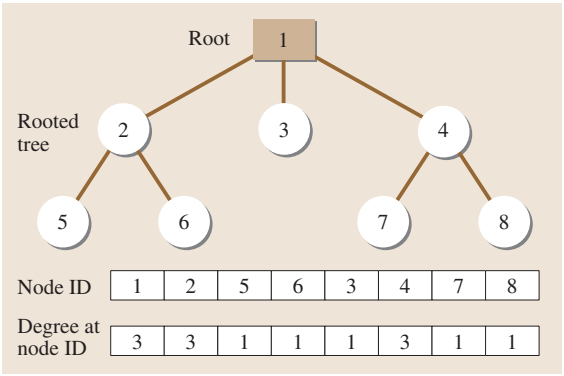


Fig. 42.18 Rooted tree and its tree-based permutation

planning is definitely 1. It is also easy to decode and evaluate. As to the initial population for an n-stage MPP problem, each individual is a permutation with $n-1$ integers whereas the integers are generated randomly with the number of all possible states in the corresponding stage.

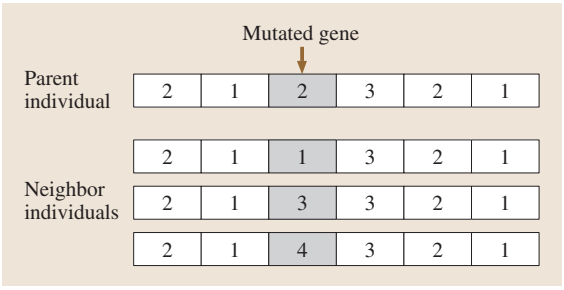


Fig. 42.19 Mutation with neighborhood search

Genetic Operation. In Zhou and Gen's method, only the mutation operation was adopted because it is easy to hybrid the neighborhood search technique to produce more adapted offspring. This hybrid mutation operation provides a great chance to evolve to the optimal solution. Figure 42.19 shows an example for this mutation operation with a neighborhood search technique supposing that the gene is at stage 3 and the number of possible states to be chosen is 4.

42.4 Scheduling Problems

Scheduling problems exist almost everywhere in real-world situations, especially in the industrial engineering world. Many scheduling problems from manufacturing industries are quite complex in nature and very difficult to solve by conventional optimization techniques.

42.4.1 Flow-Shop Sequencing Problem

The flow-shop sequencing problem is generally described as follows: there are m machines and n jobs, each job consists of m operations, and each operation requires a different machine. n jobs have to be processed in the same sequence on m machines. The processing time of job i on machine j is given by t_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, m)$. The objective is to find the sequence of jobs minimizing the maximum flow time, which is called makespan.

Heuristics for General m-Machine Problems

Genetic algorithms have been successfully applied to solve flow-shop problems. We describe Gen, Tsujimura, and Kubota's approach.

Representation. Because the flow-shop problem is essentially a permutation schedule problem [42.25–27], we can use the permutation of jobs as the representation scheme of chromosome, which is the natural representation for a sequencing problem. For example, let the k-th chromosome be

$$v_k = [3\ \ 2\ 4\ \ 1]\,,$$

meaning that the jobs sequence is $j_3,\ j_2,\ j_4,\ j_1$.

Crossover and Mutation. Here, Goldberg's PMX is used. Mutation is designed to perform random exchange; that is, it selects two genes randomly in a chromosome and exchanges their positions. An example is given in Fig. 42.20.

42.4.2 Job-Shop Scheduling

In the job-shop scheduling problem, we are given a set of jobs and a set of machines. Each machine can handle at most one job at a time. Each job consists of a chain

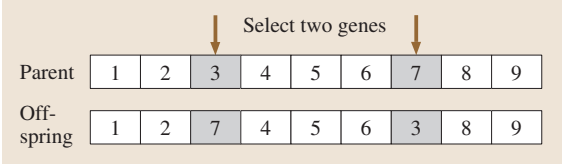


Fig. 42.20 Swap mutation

of operations, each of which needs to be processed during an uninterrupted time period of a given length on a given machine. The purpose is to find a schedule, that is, an allocation of the operations to time intervals on the machines, which has a minimum duration required to complete all jobs [42.25].

Adapted Genetic Operators

During the past two decade, various crossover operators have been proposed for literal permutation encodings, such as partial-mapped crossover (PMX), order crossover (OX), cycle crossover (CX), etc.

Partial-Mapped Crossover (PMX). PMX is explained in the previous section.

Order Crossover (OX). Order crossover was proposed by Davis. OX has the following major steps [42.14]:

Step 1. Select a substring from one parent at random.
Step 2. Produce a proto-child by copying the substring into the corresponding positions as they are in the parent.
Step 3. Delete all the symbols from the second parent that are already in the substring. The resulted sequence contains the symbols the proto-child needs.
Step 4. Place the symbols into the unfixed positions of the proto-child from left to right according to the order of the sequence to produce an offspring.

Cycle Crossover (CX). Cycle crossover was proposed by *Oliver* et al.. CX works as follows [42.25]:

Step 1. Find the cycle which is defined by the corresponding positions of symbols between parents.
Step 2. Copy the symbols in the cycle to a child with the corresponding positions of one parent.
Step 3. Determine the remaining symbols for the child by deleting those symbols which are already in the cycle from the other parent.
Step 4. Fill the child with the remaining symbols.

Mutation. It is relatively easy to make some mutation operators for the permutation representation. During the last decade, several mutation operators have been proposed for permutation representation, such as inversion, insertion, displacement, reciprocal exchange mutation, and shift mutation [42.9]. *Inversion mutation* selects two positions within a chromosome at random and then inverts the substring between these two positions. *Insertion mutation* selects a gene at random and inserts it in a random position.

42.4.3 Resource-Constrained Projected Scheduling Problem

The problem of scheduling activities under resource and precedence restrictions with the objective of minimizing the project duration is referred to as the resource-constrained project scheduling problem in the literature [42.25, 28].

The problem can be stated mathematically as follows:

$$\min\ t_n\ , \tag{42.56}$$

$$s.t.,\ t_j - t_i \geq d_i\ , \quad \forall j \in S_i\ , \tag{42.57}$$

$$\sum_{t_i \in A_{t_i}} r_{ik} \leq b_k, \quad k = 1, 2, \ldots, m\ , \tag{42.58}$$

$$t_i \geq 0, \quad i = 1, 2, \ldots, n\ , \tag{42.59}$$

where t_i is the starting time of activity i, d_i the duration (processing time) of activity i, S_i the set of successors of activity i, r_{ik} the amount of resource k required by activity i, b_k the total availability of resource k, A_{ti} the set of activities in process at time t_i, and m the number of different resource types. Activities 1 and n are dummy activities which mark the beginning and end of the project. The objective is to minimize the total project duration.

A. Priority-Based Encoding

For this problem, priority-based encoding is used; it is explained in the previous section.

B. Genetic Operators

Position-Based Crossover. The position-based crossover operator is used. This crossover is explained in the previous section.

Swap Mutation. The swap mutation operator was used here, which simply selects two positions at random and swaps their contents, as shown in Fig. 42.21.

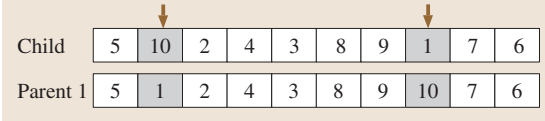


Fig. 42.21 Swap mutation operator

42.4.4 Multiprocessor Scheduling

The multiprocessor scheduling is to assign n tasks to m processors in such a way that precedence constraints are maintained, and to determine the start and finish times of each task with the objective of minimizing the completion time. There is a paper which deals with real-time tasks [42.29]. However, here we introduce an algorithm concerned with general tasks. The mathematical formulation of the problem is given as

$$\min[\max_j(x_j y_{ij})]\,, \tag{42.60}$$

$$\text{s.t.} x_k - x_j \geq p_k\,, \quad T_j \prec T_k\,, \tag{42.61}$$

$$\sum_{j=1}^{n} p_j y_{ij} \leq t_{\max}\,, \quad i = 1, \ldots, n\,, \tag{42.62}$$

$$\sum_{i=1}^{m} y_{ij} = 1\,, \quad j = 1, \ldots, m\,, \tag{42.63}$$

$$y_{ij} = 0 \;\; or \;\; 1\,, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n\,, \tag{42.64}$$

where

$$y_{ij} = \begin{cases} 1\,, & \text{if task } T_j \text{ is assigned to processor } P_i \\ 0\,, & \text{otherwise}\,, \end{cases} \tag{42.65}$$

and where $t_{\max} = \max_i(t_i)$, x_j is the completion time of task T_j, p_j is the processing time of task T_j, t_i is the time required to process all tasks assigned to process P_i, and $\prec$ represents a precedence relation; a precedence relation between tasks, $T_j \prec T_k$, means that T_k precedes T_j.

Genetic Algorithm for MSP

For the chromosome representation scheme and genetic operations, we adopt the concept of the height function [42.10], which considers precedence relations among the tasks in the implementation of a genetic algorithm.

Height Function. To facilitate the generation of the schedule and the construction of the genetic operators, we define the *height* of each task in the task graph as

$$\text{height}(T_i) = \begin{cases} 0 & \text{if pre}(T_i) = \phi \\ 1 + \max\limits_{T_j \in \text{pre}(T_i)} [\text{height}(T_j)]\,, & \text{otherwise} \end{cases} \tag{42.66}$$

$$\begin{aligned} \text{height}'(T_j) = {} & \text{rand} \in \{\max[\text{height}(T_i)] + 1, \\ & \min[\text{height}'(T_k)] - 1\} \text{ over all } T_i \in \text{pre } T_j \\ & \text{and } T_k \in \text{suc}\,(T_j)\,, \end{aligned} \tag{42.67}$$

where $\text{pre}(T_j)$ is the set of predecessors of T_j and $\text{suc}(T_j)$ is the set of successors of T_j.

Representation. The chromosome representation used here is based on the schedule of the tasks in each processor. The representation of the schedule for genetic algorithms must accommodate the precedence relations between the computational tasks.

Genetic Operators. The function of the genetic operators is to create new search nodes based on the current population of search nodes. New search nodes are typically constructed by combining or rearranging parts of the old search nodes.

Operation 1. Operation 1 is performed in the following steps

Step 1. Generate a random number c from the range $[1, \max(\text{height}')]$.
Step 2. Place the cut-point at each processor in such a way that the tasks' height$'$ before the cut-point is less than c and more than or equal to c after the cut-point.
Step 3. Exchange the second partial schedules.

Operation 2. Operation 2 is performed in the following steps

Step 1. Generate a random number c from the range $[1, \max(\text{height}')]$.
Step 2. At each processor, pick all tasks whose height$'$ is c.
Step 3. Replace the position of all tasks randomly.

42.5 Reliability Design Problem

Reliability optimization appeared in the late 1940s and was first applied to communication and transportation systems. Much of the early work was confined to the analysis of certain performance aspects of systems. One

goal of the reliability engineer is to find the best way to increase system reliability. The reliability of a system can be defined as the probability that the system has operated successfully over a specified interval of time under stated conditions.

42.5.1 Simple Genetic Algorithm for Reliability Optimization

The problem is to maximize the system reliability subject to three nonlinear constraints with parallel redundant units in subsystems that are subject to A failures, which occur when the entire subsystem is subjected to the failure condition. It can be mathematically stated as follows:

$$\max R(m) = \prod_{i=1}^{3} \left\{ 1 - [1 - (1 - q_{i1})^{m_i+1}] - \sum_{u=2}^{4} (q_{iu})^{m_i+1} \right\} , \quad (42.68)$$

$$\text{s.t.} \quad G_1(m) = (m_1+3)^2 + (m_2)^2 + (m_3)^2 \le 51 , \quad (42.69)$$

$$G_2(m) = 20 \sum_{i=1}^{3} \left[m_i + \exp(-m_i) \right] \ge 120 , \quad (42.70)$$

$$G_3(m) = 20 \sum_{i=1}^{3} \left[m_i \exp(-m_i/4) \right] \ge 65 , \quad (42.71)$$

$$1 \le m_1 \le 4, \quad 1 \le m_2, \quad m_3 \le 7 , \quad (42.72)$$

$$m_i \ge 0 : \text{integer}, \quad i = 1, 2, 3 , \quad (42.73)$$

Table 42.1 Failure modes and probabilities in each subsystem

Subsystem i	Failure modes $s_i = 4,\ h_i = 1$	Failure probability q_{iu}
1	O	0.01
	A	0.05
	A	0.10
	A	0.18
2	O	0.08
	A	0.02
	A	0.15
	A	0.12
3	O	0.04
	A	0.05
	A	0.20
	A	0.10

where $m = (m_1\, m_2\, m_3)$. The subsystems are subject to four failure modes ($s_i = 4$) with one O failure ($h_i = 1$) and three A failures, for $i = 1, 2, 3$. For each subsystem the failure probability is shown in Table 42.1.

Genetic Approach

Representation. The integer value of each variable m_i is represented as a binary string. The length of the string depends on the upper bound u_i of the redundant units. For instance, when the upper bound u_i equals 4, we need three binary bits to represent m_i. In this example, the upper bounds of the redundant units in each subsystem are $u_1 = 4$, $u_2 = 7$, $u_3 = 7$, so each decision variable m_i needs three binary bits. This means that a total of nine bits are required. If $m_1 = 2$, $m_2 = 3$, and $m_3 = 3$, we have the following chromosome:

$$\begin{aligned} \mathrm{v} &= [x_{33}\ \ x_{32}\ \ x_{31}\ \ x_{23}\ \ x_{22}\ \ x_{21}\ \ x_{13}\ \ x_{12}\ \ x_{11}] \\ &= [0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ ] \end{aligned}$$

where x_{ij} is the symbol for the j-th binary bit of variable m_i.

Crossover. One-cut-point crossover is used here.

Mutation. Mutation is performed on a bit-by-bit basis.

42.5.2 Reliability Design with Redundant Unit and Alternatives

Gen, *Yokota*, *Ida* and *Taguchi* further extended their work to the reliability optimization problem by considering both redundant units and alternative design [42.19, 30, 31].

The example used here was firstly given by *Fyffe* et al.as follows:

$$\max R(\boldsymbol{m}, \boldsymbol{\alpha}) = \prod_{i=1}^{14} \left\{ 1 - \left[1 - R_i(\alpha_i) \right]^{m_i} \right\} , \quad (42.74)$$

$$\text{s.t.} \quad G_1(\boldsymbol{m}, \boldsymbol{\alpha}) = \sum_{i=1}^{14} c_i(\alpha_i) m_i \le 130 , \quad (42.75)$$

$$G_2(\boldsymbol{m}, \boldsymbol{\alpha}) = \sum_{i=1}^{14} w_i(\alpha_i) m_i \le 170 , \quad (42.76)$$

$$1 \le m_i \le u_i , \quad \forall i , \quad (42.77)$$

$$1 \le \alpha_i \le \beta_i , \quad \forall i , \quad (42.78)$$

$$m_i , \ \alpha_i \ge 0 : \ \text{integer } \forall i , \quad (42.79)$$

where α_i represents the design alternative available for the i-th subsystem, m_i represents the identical units used

in redundancy for the ith subsystem, u_i is the upper bound of the redundant units for the i-th subsystem, and β_i is the upper bound of alternative design for the i-th subsystem.

Genetic Approach

Representation. The representation can be written as follows:

$$\boldsymbol{v}_k = [\ (\alpha_{k1}\ ,\ m_{k1})\ (\alpha_{k2}\ ,\ m_{k2})\ \cdots\ (\alpha_{k14}\ ,\ m_{k14})\ ]\ ,$$

where α_{ki} is a design alternative, m_{ki} is a redundant unit, the subscript k is the index of chromosome.

Crossover. The uniform crossover operator given by Syswerda is used here, which has been shown to be superior to traditional crossover strategies for combinatorial problem. Uniform crossover firstly generates a random crossover mask and then exchanges relative genes between parents according to the mask. A crossover mask is simply a binary string with the same size of chromosome.

42.5.3 Network Reliability Design

A communication network can be represented by an undirected graph $G = (V, E)$, in which the nodes V and edges E represent computer sites and communication cables, respectively. A graph G is connected if there is at least one path between ever pair of nodes i and j, which minimally requires a spanning tree with $(n-1)$ edges. The following notations are defined to describe the optimal design problem of all-terminal reliable networks: n is the number of nodes, $x_{ij} \in (0, 1)$ is the decision variable representing the edge between node i and node j, $\boldsymbol{x}(= \{x_{12}, x_{13}, \cdots, x_{n-1,n}\})$ is a topology architecture for the network design, x^* is the best solution found so far, p is the edge reliability for all edges, q is the edge unreliability for all edges (i. e., $p+q=1$), $R(x)$ is the all-terminal reliability of the network design x, $R_{\min}$ is the network reliability requirement, $R_U(x)$ is the upper bound on the reliability of the candidate network, c_{ij} is the cost of the edge between node i and node j, $c_{\max}$ is the maximum value of c_{ij}, δ has the value of 1 if $R(x) < R_{\min}$ and is 0 otherwise, E' is a set of operational edges ($E' \subseteq E$), Ω is all operational states (E'). The optimal design of network can be represented as follows [42.10, 32]:

$$\min Z(\boldsymbol{x}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} x_{ij}\ , \tag{42.80}$$

$$\text{s. t.} \quad R(\boldsymbol{x}) \geq R_{\min}\ . \tag{42.81}$$

Genetic Approach

Representation. A genetic algorithm lends itself to this problem because each network design $\boldsymbol{x}$ is easily formed into a binary string which can be used as a chromosome for genetic algorithms. Each element of the chromosome represents a possible edge in the network design problem, so there are $n \times (n-1)/2$ string components in each candidate architecture Z.

Crossover. The one-cut-point crossover operation is used.

Mutation. The bit-flip mutation operation is employed, performed on a bit-by-bit basis.

42.5.4 Tree-Based Network Topology Design

Consider a local-area network (LAN) that connects m users (stations). Also, we assume the $n \times n$ service center topology matrix X_1, which represents the connection between service centers. An element $\boldsymbol{x}_{1ij}$ is represented as

$$\boldsymbol{x}_{1ij} = \begin{cases} 1\ , & \text{if the centers } i \text{ and } j \text{ are connected} \\ 0\ , & \text{otherwise}\ . \end{cases} \tag{42.82}$$

Assume that the LAN is partitioned into n segments (service centers or clusters). The users are distributed over those n service centers. The $n \times m$ clustering matrix $\boldsymbol{X}_2$ specifies which user belongs to which center. Thus

$$\boldsymbol{x}_{2ij} = \begin{cases} 1\ , & \text{if user } j \text{ belongs to center } i \\ 0\ , & \text{otherwise}\ . \end{cases} \tag{42.83}$$

A user can only belong to one center; thus, $\forall j = 1, 2, \ldots, m$, $\sum_{i=1}^{n} x_{2ij} = 1$. We define an $n \times (n+m)$ matrix $\boldsymbol{X}$ called the spanning tree matrix ($[\boldsymbol{X}_1\ \ \boldsymbol{X}_2]$). The bicriteria LAN topology design problem can be formulated as the following nonlinear 0–1 programming model [42.10, 33, 34]:

$$\max R(\boldsymbol{X})\ , \tag{42.84}$$

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{1ij} x_{1ij} + \sum_{i=1}^{n} \sum_{j=1}^{m} w_{2ij} x_{2ij}\ , \tag{42.85}$$

$$\text{s. t.} \sum_{j=1}^{m} x_{1ij} \leq g_i\ , \quad i = 1, 2, \ldots, n\ , \tag{42.86}$$

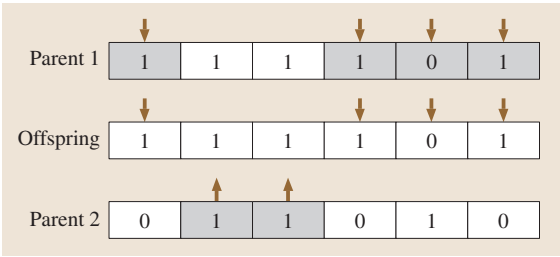


Fig. 42.22 Uniform crossover operator

$$\sum_{i=1}^{n} x_{2ij} = 1\,, \quad j = 1, 2, \ldots, m\,, \tag{42.87}$$

where $R(\boldsymbol{X})$ is the network reliability, w_{1ij} is the weight of the link between the centers i and j, w_{2ij} is the weight of the link between the center i and the user j, g_i is the maximum number that can connect to the center i.

Genetic Approach

Representation. We can easily construct an encoding as follows:

procedure: Encoding of Prüfer number

Step 1. Let node i be the smallest labeled leaf node in a labeled tree T.
Step 2. Let j be the first digit in the encoding as the node j incident to node i is uniquely determined. The encoding is built by appending digits from left to right.
Step 3. Remove node i and the link from i to j; thus we have a tree with $k-1$ nodes.
Step 4. Repeat the above steps until one link is left. We produce a Prüfer number or an encoding with $k-2$ digits between 1 and k inclusive.

Crossover. Uniform crossover is used. This type of crossover is accomplished by selecting two parent solutions and randomly taking a component from one parent to form the corresponding component of the offspring Fig. 42.22.

Mutation. Swap mutation is used, as explained in the previous section.

42.6 Logistic Network Problems

The transportation problem is a basic model in the logistic networks. Many scholars have since refined and extended the basic transportation model to include not only the determination of optimum transportation patterns but also the analysis of production scheduling problems, transshipment problems, and assignment problems.

42.6.1 Linear Transportation Problem

The linear transportation problem (LTP) involves the shipment of some homogeneous commodity from various origins or sources of supply to a set of destinations, each demanding specified levels of the commodity. The usual objective function is to minimize the total transportation cost or total weighted distance or to maximize the total profit contribution from the allocation [42.35].

Given m origins and n destinations, the transportation problem can be formulated as a linear programming model:

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}\,, \tag{42.88}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1, 2, \ldots, m\,, \tag{42.89}$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j, \quad j = 1, 2, \ldots, n\,, \tag{42.90}$$

$$x_{ij} \ge 0, \quad \text{for all } i \text{ and } j\,, \tag{42.91}$$

where x_{ij} is the amount of units shipped from origin i to destination j; c_{ij} is the cost of shipping one unit from source i to destination j; a_i is the number of units available at origin i; and b_j is the number of units demanded at destination j.

Genetic Approach

Representation. Perhaps the matrix is the most natural representation of a solution for the transportation problem. The allocation matrix for the transportation problem can be written as follows:

$$\boldsymbol{X}_p = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \tag{42.92}$$

where $\boldsymbol{X}_p$ denotes the p-th chromosome and x_{ij} is the corresponding decision variable.

Crossover. Assume that two matrices $\boldsymbol{X}_1 = (x_{ij}^1)$ and $\boldsymbol{X}_2 = (x_{ij}^2)$ are selected as parents for the crossover operation. The crossover is performed in the following three main steps:

Step 1. Create two temporary matrices $\boldsymbol{D} = (d_{ij})$ and $\boldsymbol{R} = (r_{ij})$ as follows: $d_{ij} = \left[\left(x_{ij}^1 + x_{ij}^2\right)/2\right]$ and $r_{ij} = \left(x_{ij}^1 + x_{ij}^2\right) \mod 2$.

Step 2. Divide matrix $\boldsymbol{R}$ into two matrices $\boldsymbol{R}^1 = \left(r_{ij}^1\right)$ and $\boldsymbol{R}^2 = \left(r_{ij}^2\right)$ such that: $\boldsymbol{R} = \boldsymbol{R}^1 + \boldsymbol{R}^2$

Step 3. Then we produce two offspring of $\boldsymbol{X}_1'$ and $\boldsymbol{X}_2'$ as follows: $\boldsymbol{X}_1' = \boldsymbol{D} + \boldsymbol{R}^1$ and $\boldsymbol{X}_2' = \boldsymbol{D} + \boldsymbol{R}^2$

Mutation. The mutation is performed in following three main steps:

Step 1. Make a submatrix from the parent matrix. Randomly select $\{i_1, \cdots, i_p\}$ rows and $\{j_1, \cdots, j_q\}$ columns to create a $(p * q)$ submatrix $\boldsymbol{Y} = (y_{ij})$, where $\{i_1, \cdots, i_p\}$ is a proper subset of $\{1, 2, \ldots, m\}$ and $2 \le p \le m$, $\{j_1, \cdots, j_q\}$ is a proper subset of $\{1, 2, \ldots, n\}$ and $2 \le q \le n$, and y_{ij} takes the value of the element in the crossing position of selected row i and column j in the parent matrix.

Step 2. Reallocate commodity for the submatrix. The available amount of commodity a_i^y and the demands b_j^y for the submatrix are determined as follows:

$$a_i^y = \sum_{j \in \{j_1, \cdots, j_q\}} y_{ij}\,, \quad i = i_1, i_2, \cdots, i_p\,, \tag{42.93}$$

$$b_j^y = \sum_{i \in \{i_1, \cdots, i_p\}}^{y_{ij}}\,, \quad j = j_1, j_2, \cdots, j_q\,. \tag{42.94}$$

Step 3. Replace appropriate elements of the parent matrix by new elements from the reallocated submatrix $\boldsymbol{Y}$.

Spanning Tree-Based Approach. Transportation problems (TP) as a special type of network problem have a special data structure characterized as a transportation graph in their solutions. The spanning tree-based GA incorporating this data structure of TP was proposed by Gen and Li. This GA utilized the Prüfer number encoding based on a spanning tree, which is adopted because it is capable of representing all possible trees. Using the Prüfer number representation the memory only requires $m + n - 2$ entries for a chromosome in the TP. Transportation problems have separable sets of nodes for plants and warehouses. From this point, *Gen* and *Cheng* designed the criterion for feasibility of the chromosome. The proposed spanning tree-based GA can find the optimal or near-optimal solution for transportation problems in the solution space [42.10].

42.6.2 Multiobjective Transportation Problem

In the transportation problem, multiple objectives are required in practical situations, such as minimizing transportation cost, minimizing the average shipping time to priority customers, maximizing production using a given process, minimizing fuel consumption, and so on. The traditional multiobjective transportation problem (mTP) with m plants and n warehouses can be formulated as

$$\min z_q = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^q x_{ij} \quad q = 1, 2, \ldots, Q\,, \tag{42.95}$$

$$\text{s. t.} \sum_{j=1}^{n} x_{ij} \le a_i\,, \quad i = 1, 2, \ldots, m\,, \tag{42.96}$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j, \quad j = 1, 2, \ldots, n\,, \tag{42.97}$$

$$x_{ij} \ge 0\,, \quad \forall i,\ j\,, \tag{42.98}$$

where q means the q-th objective function.

Spanning Tree-based GA for Multi-objective TP

The Pareto optimal solutions are usually characterized as the solutions of the multiobjective programming problem [42.36, 37].

42.6.3 Bicriteria Transportation Problem with Fuzzy Coefficients

Consider the following two objectives: minimizing total transportation cost and minimizing total delivery time. Let $\tilde{c}_{ij}^1$ be the fuzzy data representing the transportation cost of shipping one unit from plant i to warehouse j, let $\tilde{c}_{ij}^2$ be the fuzzy data representing the delivery time of shipping one unit of the product from plant i to warehouse j, a_i be the number of units available at plant i,

and b_j be the number of units demanded at warehouse j. This problem with m plants and n warehouses can be formulated as [42.10]:

$$\min \tilde{z}_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} \tilde{c}_{ij}^1 x_{ij} \,, \tag{42.99}$$

$$\min \tilde{z}_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \tilde{c}_{ij}^2 x_{ij} \,, \tag{42.100}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij} \le a_i \,, \quad i = 1, 2, \ldots, m \,, \tag{42.101}$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j \,, \quad j = 1, 2, \ldots, n \,, \tag{42.102}$$

$$x_{ij} \ge 0 \,, \quad \forall i \,, \; j \,, \tag{42.103}$$

where x_{ij} is the unknown quantity to be transported from plant i to warehouse j.

Genetic Approach

The proposed genetic algorithm approach is based on spanning tree. In multicriteria optimization, we are interested in finding Pareto solutions. When the coefficients of objectives are represented with fuzzy numbers, the objective values become fuzzy numbers. Since a fuzzy number represents many possible real numbers, it is not easy to compare solutions to determine which is the Pareto solution. Fuzzy ranking techniques can help us to compare fuzzy numbers. In this approach, Pareto solutions are determined based on the ranked values of fuzzy objective functions, and genetic algorithms are used to search for Pareto solutions.

Representation. The spanning-tree encoding, the Prüfer number, is used to represent the candidate solution. The criterion for the solution's feasibility designed in the proposed spanning-tree-based GA is also employed.

Crossover. For simplicity one-point crossover is used.

Mutation. Inversion mutation and displacement mutation are used.

42.6.4 Supply-Chain Management (SCM) Network Design

Supply-chain management (SCM) aims to choose the subset of plants and distribution centers to be opened and to design the distribution network strategy that can satisfy all capacities and demand requirements imposed by customers with minimum cost. We formulate the problem by using the following mixed integer linear programming model (MILP) [42.28, 38–41]:

$$\min \sum_i \sum_j s_{ij} x_{ij} + \sum_j \sum_k t_{jk} y_{jk} + \sum_k \sum_l u_{kl} z_{kl} + \sum_j f_j w_j + \sum_k g_k z_k \tag{42.104}$$

$$\text{s.t.} \sum_j x_{ij} \le a_i, \; \forall i \,, \tag{42.105}$$

$$\sum_k y_{jk} \le b_j w_j \,, \; \forall j \,, \tag{42.106}$$

$$\sum_j w_j \le P \,, \tag{42.107}$$

$$\sum_l z_{kl} \le c_k z_k, \; \forall k \,, \tag{42.108}$$

$$\sum_k z_k \le W \,, \tag{42.109}$$

$$\sum_k z_{kl} \ge d_l, \; \forall l \,, \tag{42.110}$$

$$w_j, z_k = (0, 1) \,, \; \forall j, \; k \,, \tag{42.111}$$

$$x_{ij}, y_{jk}, z_{kl} \ge 0, \; \forall i, \; j, \; k, \; l \,, \tag{42.112}$$

where i is the number of suppliers, j is the number of plants, K is the number of distribution centers, L is the number of customers, a_i is the capacity of supplier i, b_i is the capacity of plant j, c_k is the capacity of distribution center k, d_l is the demand of customer l, s_{ij} is the unit cost of production in plant j using material from supplier i, t_{jk} is the unit cost of transportation from plant j to the distribution center k, u_{kl} is the unit cost of transportation from distribution k to customer l, f_j is the fixed cost for operating plant j, g_k is the fixed cost for operating distribution center k, W is an upper limit on the total number of distribution centers that can be opened and P is an upper limit on the total number of plants that can be opened.

Here, x_{ij} is the quantity produced at plant j using raw material from supplier i, y_{jk} is the amount shipped from plant j to distribution center k and z_{kl} is the amount shipped from distribution center k to customer l. w_j and z_k are defined as

$$w_j = \begin{cases} 1, & \text{if production takes place at plant } j \\ 0, & \text{otherwise} \,, \end{cases} \tag{42.113}$$

$$z_k = \begin{cases} 1, & \text{if distribution center } k \text{ is opened} \\ 0, & \text{otherwise} . \end{cases} \tag{42.114}$$

Genetic Approach

Crossover. The crossover is done by exchanging the information of two parents to provide a powerful exploration capability. We employ a one-cut-point crossover operation, which randomly selects one cut-point and exchanges the right parts of the two parents to generate offspring.

Mutation. Modifying one or more of the gene values of an existing individual, mutation creates a new individual to increase the variability of the population. We use inversion and displacement mutation operations.

42.7 Location and Allocation Problems

Location–allocation problems arise in many practical settings. The classical single location–allocation problem is to find the single location which minimizes the summed distance from some number of fixed points, representing customers with known locations.

42.7.1 Location–Allocation Problem

There are m facilities to be located, and n customers with known locations are to be allocated to the variable facilities. Each customer has the requirement q_j, $j = 1, 2, \ldots, n$, and each facility has the capacity b_i, $i = 1, 2, \ldots, m$. We need to find the locations of facilities and allocations of customers to facilities so that the total summed distance among the customers and their serving facilities is minimized Fig. 42.23. This problem is formulated mathematically as [42.9]:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} \sqrt{(x_i - u_j)^2 + (y_i - v_j)^2}\ z_{ij} \tag{42.115}$$

$$\text{s.t.} \sum_{j=1}^{n} q_j \cdot z_{ij} \le b_i , \quad i = 1, 2, \ldots, m , \tag{42.116}$$

$$\sum_{i=1}^{m} z_{ij} = 1 , \quad j = 1, 2, \ldots, n , \tag{42.117}$$

$$z_{ij} = 0 \text{ or } 1 , \ i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n , \tag{42.118}$$

where

$$(u_j, v_j) = \text{location of customer } j, \ j = 1, 2, \ldots, n , \tag{42.119}$$

$$(x_j, y_j) = \text{location of facility } i, \tag{42.120}$$

$$\text{decision variables } i = 1, 2, \ldots, m , \tag{42.121}$$

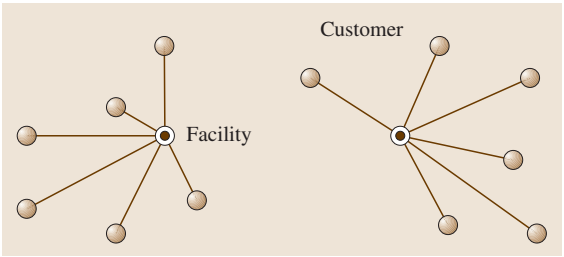


Fig. 42.23 Location–allocation problem

$$z_{ij} = 0 - 1 \text{ decision variable} ,$$

$$z_{ij} = \begin{cases} 1, & \text{customer } j \text{ is served by facility } i \\ 0, & \text{otherwise} \end{cases} . \tag{42.122}$$

A. Genetic Approach

Representation. Since location variables are continuous, the float-value chromosome representation is used. A chromosome is given as follows:

$$\boldsymbol{c}^k = \left[\left(x_1^k, y_1^k\right) \left(x_2^k, y_2^k\right) \cdots \left(x_m^k, y_m^k\right) \right]$$

where $\left(x_i^k, y_i^k\right)$ is the location of the i-th facility in the k-th chromosome, $i = 1, 2, \ldots, m$.

Crossover. Two mating strategies are used: one is free mating, which selects two parents at random; another is dominating mating, which uses the fittest individual as a fixed parent and randomly selects another parent from the population pool. These two strategies are used alternatively in the evolutionary process.

Suppose two parents with the following chromosomes are selected to produce a child

$$\boldsymbol{c}^{k_1} = \left[\left(x_1^{k_1}, y_1^{k_1}\right) \left(x_2^{k_1}, y_2^{k_1}\right) \cdots \left(x_m^{k_1}, y_m^{k_1}\right) \right] ,$$

$$\boldsymbol{c}^{k_2} = \left[\left(x_1^{k_2}, y_1^{k_2}\right)\left(x_2^{k_2}, y_2^{k_2}\right) \cdots \left(x_m^{k_2}, y_m^{k_2}\right) \right] .$$

Only one child is allowed to be produced:

$$\begin{aligned} \boldsymbol{c} &= [(x_1, y_1)(x_2, y_2) \cdots (x_m, y_m)] , \\ x_i &= r_i \cdot x_i^{k_1} + (1 - r_i) \cdot x_i^{k_2} , \\ y_i &= r_i \cdot y_i^{k_1} + (1 - r_i) \cdot y_i^{k_2} . \end{aligned} \tag{42.123}$$

Mutation. Suppose the candidate chromosome to be mutated is as follows:

$$\boldsymbol{c}^k = \left[\left(x_1^k, y_1^k\right)\left(x_2^k, y_2^k\right) \cdots \left(x_m^k, y_m^k\right) \right]$$

then the chromosome of the child produced by subtle mutation $\boldsymbol{c} = [x_1, y_1, x_2, y_2, \cdots, x_m, y_m]$ is as follows:

$$\begin{aligned} x_i &= x_i^k + \text{random value in } [-\varepsilon, \varepsilon] , \\ y_i &= y_i^k + \text{random value in } [-\varepsilon, \varepsilon] . \end{aligned} \tag{42.124}$$

B. Numerical Example

Cooper and *Rosing*'s examples are used to test the effectiveness of this method [42.42]. Cooper carefully constructed the front half data which contains three natural groups and Rosing increased the number of customers with random points. These examples provide a good benchmark to test the effectiveness of the proposed method because their global optimal solutions have already been found.

These examples include 30 customers whose location coordinates are shown in Table 42.2. Theirs is a common location–allocation problem where the requirements of the customers are treated as equal and the capacities of the facilities are assumed to be unlimited.

Both the alternative location-allocation (ALA) method and the hybrid evolutionary method (HEM) were applied to solve these examples. When using the ALA method, it was run to solve the same problem 40 times from randomly generated initial locations. The computed results are given in Table 42.3 [42.10]. In the table, the percent error was calculated by (actual value−optimal value)/optimal value ×100%.

Table 42.2 Coordinates of Cooper and Rosing's example

Order number	X	Y	Order number	X	Y
1	5	9	16	53	8
2	5	24	17	1	34
3	5	48	18	33	8
4	13	4	19	3	26
5	12	19	20	17	9
6	13	39	21	53	20
7	28	37	22	24	17
8	21	45	23	40	22
9	25	50	24	22	41
10	31	9	25	7	13
11	39	2	26	5	17
12	39	16	27	39	3
13	45	22	28	50	50
14	41	30	29	16	40
15	49	31	30	22	45

42.7.2 Capacitated Plant Location Problem

The capacitated plant location problem (cPLP) is referred to as a fixed-charge problem to determine the locations of plants with minimal total cost, including production, shipping costs, and fixed costs where

Table 42.3 Comparison results of Cooper and Rosing's example

Problem n/m	Rosing's method optimal objective	ALA Best	ALA Percent error	HEM Best	HEM Percent error
15/2	214.281	219.2595	2.32	214.2843	0.0015
15/3	143.197	144.8724	1.17	143.2058	0.0061
15/4	113.568	115.4588	1.69	113.5887	0.0182
15/5	97.289	99.4237	2.19	97.5656	0.2843
15/6	81.264	84.0772	3.46	83.0065	2.14
30/2	447.728	450.3931	0.5952	447.73	0.0004
30/3	307.372	310.3160	0.9578	307.3743	0.0007
30/4	254.148	258.4713	1.7010	254.2246	0.0301
30/5	220.057	226.8971	3.1083	220.4335	0.1711
30/6	–	208.4301	3.4940	201.4031	0.0

the plants are located. In this case, m sources (or facility locations) produce a single commodity for n customers, each with demand of b_j ($j = 1, \ldots, n$) units. If a particular source i is opened (or facility is built), it has a fixed cost $d_i \geq 0$ and a production capacity $a_i \geq 0$ associated with it. There is also a positive cost c_{ij} for shipping a unit from source i to customer j. The problem is to determine the locations of the plants so that capacities are not exceeded and demands are met, all at a minimal total cost. The cPLP is a mixed integer program, as shown in the following [42.10]

$$\min z(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} d_i y_i \qquad (42.125)$$

$$\text{s.t.} \sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n, \qquad (42.126)$$

$$\sum_{j=1}^{n} x_{ij} \leq a_i y_i, \quad i = 1, 2, \ldots, m, \qquad (42.127)$$

$$x_{ij} \geq 0, \quad \forall i, j, \qquad (42.128)$$

$$y_i = 0 \text{ or } 1, \quad i = 1, 2, \ldots, m. \qquad (42.129)$$

The variables are x_{ij} and y_i, which represent the amount shipped from plant i to warehouse j and whether a plant is open (or located) ($y_i = 1$) or closed ($y_i = 0$), respectively.

Spanning Tree-Based GA for Plant Location Problems

The spanning tree-based GA for the capacitated plant location problem is the same as that of the fixed-charge transportation problem except there is a different evaluation function in the evolutionary process.

42.7.3 Obstacle Location–Allocation Problem

There are n customers with known locations and m facilities to be built to supply some kind of services to all customers, for example, supplying materials or energy. There are also p obstacles representing some forbidden areas. The formulation of the mathematical model is based on the following assumptions:

- customer j has service demand q_j, $j = 1, 2, \ldots, n$,
- facility i has service capacity b_i, $i = 1, 2, \ldots, m$,
- each customer should be served by only one facility,
- new facilities should not be built within any obstacle,
- connecting paths between facilities and customers should not be allowed to pass through any of the obstacles.

The problem is to choose the best locations for facilities so that the sum of distances between customers and their serving facilities is minimal, as illustrated in Fig. 42.24. The obstacle location–allocation problem can be formulated as follows [42.9]:

$$\min f(\mathrm{D}, z) = \sum_{i=1}^{m}\sum_{j=1}^{n} t(\mathrm{D}_i, \mathrm{C}_j)\cdot z_{ij} \qquad (42.130)$$

$$\text{s.t.} \sum_{j=1}^{n} d_j z_{ij} \leq q_i, \quad i = 1, 2, \cdots, m, \qquad (42.131)$$

$$\sum_{i=1}^{m} z_{ij} = 1, \quad j = 1, 2, \cdots, n, \qquad (42.132)$$

$$\mathrm{D}_i = (x_i, y_i) \notin \mathrm{Q}_k, \quad i = 1, 2, \cdots, m, k = 1, 2, \cdots, q, \qquad (42.133)$$

$$(x_i, y_i) \in \mathrm{R}_\mathrm{T}, \quad i = 1, 2, \cdots, m, \qquad (42.134)$$

$$x_i, y_j \in R \quad i = 1, 2, \cdots, m \qquad (42.135)$$

$$z_{ij} = 1 \text{ or } 0, \quad i = 1, 2, \cdots, m, j = 1, 2, \cdots, n, \qquad (42.136)$$

where $\mathrm{C}_j = (u_j, v_j)$ is the location of the j-th customer, $\mathrm{D}_i = (x_i, y_i)$ is the decision variable, the location of the j-th distribution center DC_i should not fall within any of the obstacles, $t(\mathrm{D}_i,\mathrm{C}_j)$ is the shortest connecting path from the set of possible paths between the distribution center DC_i and the customer C_j which avoids all obstacles, R_T is the total area considered for the location and allocation problem and z_{ij} is a 0–1 decision variable; $z_{ij} = 1$ indicates that the j-th customer is served by DC_i, $z_{ij} = 0$ otherwise.

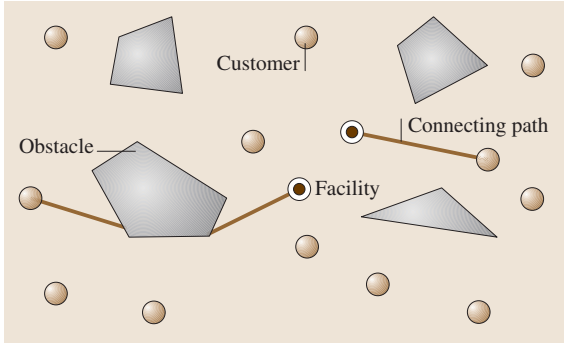


Fig. 42.24 Obstacle location–allocation problem

Hybrid Evolutionary Method

Since there are obstacles, the locations of the chromosome produced by initialization, crossover and mutation procedure may become infeasible. Generally, there are three kinds of methods to treat infeasible chromosomes. The first is to discard it, but according to the experience of other researchers this method may lead to very low efficiency. The second is to add a penalty to infeasible chromosomes. The third is to repair the infeasible chromosome according to the characteristics of the specified problem.

References

42.1 D. Fogel, A. Ghozeil: Using Fitness Distributions to Design More Efficient Evolutionary Computations, Proc. of the Third IEEE conference on Evolutionary Computation, Nagoya 1996, ed. by D. Fogel (IEEE Press, Nagoya 1996) 11–19

42.2 M. Gen, R. Cheng: Evolutionary network design: Hybrid genetic algorithms approach, Inter. J. Comp. Intell. & Appl. **3**, 357–380 (2003)

42.3 Y. Yun, M. Gen (Eds.): Adaptive hybrid genetic algorithm with fuzzy logic controller. In: *Fuzzy Sets Based Heuristics for Optimization*, ed. by J. L. Verdegay (Springer, New York 2003) pp. 251–263

42.4 C. Y. Lee, Y. S. Yun, M. Gen: Reliability optimization design for complex systems by hybrid GA with fuzzy logic controller and local search, IEICE Trans. Electr. **E85-A**, 880–891 (2002)

42.5 H. Xu, G. Vukovich: Fuzzy Evolutionary Algorithms, Automatic Robot Trajectory Generation, Proc. of the First IEEE Conference on Evolutionary Computation, Orlando 1998, ed. by D. Fogel (IEEE Press, Piscataway 1998) 595–600

42.6 H. Ishii, H. Shiode, T. Murata: A multiobjective genetic local search algorithm and its application to flowshop scheduling, IEEE Trans. Syst. Man Cyber. **28**, 392–403 (1998)

42.7 M. Gen, J. R. Kim: GA-based Optimization of Reliability Design. In: *Evolutionary Design by Computers*, ed. by P. Bentley (Morgan Kaufman, San Francisco 1999) pp. 191–218

42.8 S. Martello, P. Toth: *Knapsack Problems: Algorithms and Computer Implementations* (Wiley, Chichester 1990)

42.9 M. Gen, R. Cheng: *Genetic Algorithms & Engineering Design* (Wiley, New York 1997)

42.10 M. Gen, R. Cheng: *Genetic Algorithms & Engineering Optimization* (Wiley, New York 2000)

42.11 L. Lin, M. Gen: Node-based genetic algorithm for communication spanning tree problem, IEICE Trans. on Comm. **E89-B**(4), 1091–1098 (2006)

42.12 E. Falkenauer: Tapping the full power of genetic algorithms through suitable representation, local optimization: Application to bin packing. In: *Evolutionary Algorithms in Management Applications*, ed. by J. Biethahn, V. Nissen (Springer, Berlin Heidelberg New York 1995) pp. 167–182

42.13 L. Davis (Ed.): *Handbook of Genetic Algorithms* (Van Nostrand Reinhold, New York 1991)

42.14 Z. Michalewicz: *Genetic Algorithm + Data Structure = Evolution Programs*, 2nd edn. (Springer, New York 1994)

42.15 D. Goldberg, R. Lingle: Loci and The Traveling Salesman Problem, Proc. of the First International Conference on Genetic Algorithms, New Jersey 1985, ed. by J. Grefenstette (Lawrence Erlbaum Associates, Hillsdale 1985) 154–192

42.16 R. Cheng, M. Gen: Evolution Program for Resource Constrained Project Scheduling Problem, Proc. of the First IEEE Conference on Evolutionary Computation, Orlando 1994, ed. by D. Fogel (IEEE Press, Orlando 1994) 736–741

42.17 M. Gen, A. Kumar, J. R. Kim: Recent network design techniques using evolutionary algorithms, Int. J. Prod. Econ. **98**(2), 251.261 (2005)

42.18 R. Cheng, M. Gen: Resource constrained project scheduling problem using genetic algorithm, Inter. J. Intell. Autom. Soft Comp. **3**, 273–286 (1997)

42.19 T. Yokota, M. Gen, K. Ida, T. Taguchi: Optimal design of system reliability by an approved genetic algorithm, Trans. Inst. Electron. Inf. Commun. Eng. **J78A**, 702–209 (1995)

42.20 M. Gen, L. Lin: A new approach for shortest path routing problem by random key-based GA, Genetic and Evol. Comp. Conf., Seattle (2006)

42.21 G. Syswerda: Scheduling Optimization Using Genetic Algorithms. In: *Handbook of Genetic Algorithms*, ed. by L. Davis (Van Nostrand Reinhold, NewYork 1991) pp. 332–349

42.22 M. Gen, L. Lin, R. Cheng: Bicriteria network optimization problem using priority-based genetic algorithm, IEEJ Trans. Elect. Inf. Sys. **124**, 1972–1978 (2004)

42.23 L. Lin, M. Gen: Bicriteria network design problem using interactive adaptive-weight GA and priority-based encoding method, IEEE Trans. Evol. Comput (in reviewing)

42.24 M. Gen, L. Lin: Multi-objective hybrid genetic algorithm for bicriteria network design problem, Compl. Int. **11**, 73–83 (2005)

42.25 C. Cheng, V. Vempati, N. Aljaber: An application of genetic algorithms for flow shop problems, Eur. J. Oper. Res. **80**, 389–396 (1995)

42.26 H. Ishibuchi, N. Yamamoto, T. Murata, H. Tanaka: Genetic algorithms and neighborhood search algorithms for fuzzy flowshop scheduling problems, Fuzzy Sets and Systems **67**, 81–100 (1994)

42.27 K. Baker: *Introduction to Sequencing and Scheduling* (Wiley, New York 1974)

42.28 M. Gen, K. W. Kim, G. Yamazaki: Project scheduling using hybrid genetic algorithm with fuzzy logic controller in scm environment, J. Tsinghua Sci. Technol. **8**, 1, 19–29 (2003)

42.29 M. Yoo, M. Gen: Scheduling algorithm for real-time tasks using multiobjective hybrid genetic algorithm in heterogeneous multiprocessors system, Comp. Oper. Res. in press

42.30 G. Syswerda: Uniform Crossover in Genetic Algorithm, Proc. of the 3rd International Conference on Genetic Algorithms, San Francisco 1989, ed. by J. Schaffer (Morgan Kaufmann, San Francisco 1989) 2–9

42.31 M. Gen, K. Ida, T. Taguchi: System Reliability Optimization with Several Failure Modes by Genetic Algorithm, Proceedings of International Conference on Computers & Industrial Engineering, Japan, Ashikaga 1994, ed. by M. Gen, Ashikaga 1994) 349–351

42.32 L. Lin, M. Gen: A self control genetic algorithm for reliable communication network design, Proc. of IEEE Congress on Evol. Comp. Vancouver, IEEE Press (2006)

42.33 G. Zhou, M. Gen: A genetic algorithm approach on tree-like telecommunication network design problem, J. Oper. Res. Soc. **54**, 248–254 (2003)

42.34 A. Syarif, M. Gen: Solving exclusionary side constrained transportation problem by using a aybrid spanning tree-based genetic algorithm, J. Intell. Manuf. **14**, 389–399 (2003)

42.35 F. Budnick, D. McLeavey, R. Mojena: *Principles of Research for Management*, 2nd edn. (Irwin, Homewood 1998)

42.36 M. Gen, Y. Li: Solving Multi-Objective Transportation Problem by Spanning Tree-Based Genetic Algorithm. In: *Adaptive Computing in Design and Manufacture*, ed. by I. Parmee (Springer, New York 1998) pp. 95–108

42.37 M. Gen, A. Syalif: Hybrid genetic algorithm for multi-time period production/distribution planning, Comp. Ind. Eng. **48**(4), 799–809 (2005)

42.38 A. Syarif, Y. S. Yun, M. Gen: Study on multi-stage logistics chain network: A spanning tree-based genetic algorithm approach, Comp. Ind. Eng. **43**, 299–314 (2002)

42.39 M. Gen, A. Syarif: Multi-stage Supply Chain Network by Hybrid Genetic Algorithms with Fuzzy Logic Controller. In: *Fuzzy Sets based Heuristics for Optimization*, ed. by J. L. Verdegay (Springer, New York 2003) pp. 181–196

42.40 G. Zhou, H. Min, M. Gen: A genetic algorithm approach to the bi-criteria allocation of customers to warehouses, Int. J. Prod. Econ. **86**, 35–45 (2003)

42.41 M. Gen, F. Altiparmak, L. Lin: A genetic algorithm for two-stage transportation problem using priority-based encoding, OR Spectrum (2006)

42.42 K. Rosing: An optimal method for solving (generalized) multi-weber problem, Eur. J. Oper. Res. **58**, 414–426 (1992)

43. Scan Statistics

Section 43.1 introduces the concept of scan statistics and overviews types used to localize unusual clusters in continuous time or space, in sequences of trials or on a lattice. Section 43.2 focuses on scan statistics in one dimension. Sections 43.2.2 and 43.2.3 deal with clusters of events in continuous time. Sections 43.2.4 and 43.2.5 deal with success clusters in a sequence of discrete binary (s-f) trials. Sections 43.2.6 and 43.2.7 deal with the case where events occur in continuous time, but where we can only scan a discrete set of positions. Different approaches are used to review data when looking for clusters (the retrospective case in Sects. 43.2.2, 43.2.5, 43.2.6), and for ongoing surveillance that monitors unusual clusters (the prospective case in Sects. 43.2.2, 43.2.3, 43.2.7). Section 43.2.7 describes statistics used to scan for clustering on a circle (are certain times of the day or year more likely to have accidents?). Section 43.3 describes statistics used to scan continuous space or a two-dimensional lattice for unusual clusters. Sections 43.2 and 43.3 focus on how unusual the largest number of events within a scanning window is. Section 43.4.1 deals with scanning for unusually sparse regions. In some cases the researcher is more interested in the number of clusters, rather than the size of the largest or smallest, and Sect. 43.4.2 describes results useful for this case. The double-scan statistic of Sect. 43.4.3 allows the researcher to test for unusual simultaneous or lagged clustering of two different types of events. Section 43.4.4 describes scan statistics that can be used on data with a complex structure.

43.1 Overview

During design, monitoring, or analysis work, engineers and other scientists often need to take into account unusually large *clusters* of events in time or space. Mechanical engineers design system capacity to provide reliability to pipeline systems. Telecommunication experts seek to avoid outage caused by too many mobiles transmitting within the same area served by a single base station. Quality control experts monitor for clusters of defectives. Epidemiologists investigate *hotspots* of cancer cases, and carry out syndrome surveillance to monitor for bioterrorism attacks. Computer scientists base an information flow control mechanism on a large enough number of information packets within a temporal *sliding window*. Astronomers scan for muon clusters. Electrical engineers build in multiple redundancies to improve reliability, and use clusters of successes

as a criteria for start-up reliability. Molecular biologists search for clusters of specific types of patterns in protein or DNA sequences to focus on regions with important biologic functions. These scientists seek to determine which clusters are unlikely to occur by chance. The distributions given by various cluster statistics are tools that answer this question.

Scan statistics measure an unusually large cluster of events in time and or space. Given events distributed over a time period $(0, T)$, S_w is the largest number of events in any subinterval of length w. S_w is called the (temporal) *scan statistic*, from the viewpoint that one scans the time period $(0, T)$ with a window of size w, and finds the maximum cluster of points. A simple example illustrates this.

Example 1: A public health officer reviewing the records for a nursing home observed 60 deaths over the five-year period from January 1, 2000 through December 31, 2004. This was about average for such facilities. However, the officer observed that in the one-year period between April 1, 2003 to March 31, 2004 there were 23 deaths. Given that the 60 dates of death were independent, and occurred at random over the five-year period, how likely is it that there would be any one year period with 23 or more deaths? That is, if we scan the $T = 5$ year period with a $w = 1$ year period, how likely is it that the scan statistic $S_w \geq 23$? Note that the officer did not just divide the five-year period up into five calendar years and look at the calendar year with the largest number of deaths. In fact, the cluster observed did not occur in a calendar year. The scan statistic takes into account that the officer looked at a very large number of overlapping one-year periods. We see below that, even taking into account these multiple comparisons, the cluster of 23 deaths in a one-year period is fairly unusual. $P(S_w \geq 23, \text{ given } N = 60, w/T = 0.2) < 0.03$.

There is a large volume of literature and much current research being done on scan statistics. Three recent books [43.1–3] summarize and reference many results. Chapters 9 through 12 of [43.1] deals with scan statistics in a sequence of trials. There are useful engineering applications to reliability in consecutive systems, quality control, matching in genetic sequences, sooner and later waiting time problems and 556 references on runs and scans. Recent advances from articles by researchers are detailed in [43.2] together with applications. In [43.3], the first six chapters systematically show many applications with useful simple formulae for scan statistics, and are aimed at practitioners; the remaining 12 chapters develop the theory and methodology of scan statistics, and this is followed by a bibliography of over 600 references. In this review we draw heavily on these references, particularly [43.3], and subsequent research to give an overview of scan statistics, and highlight many results that have proved useful in many scientific applications. Scan statistics have been developed and applied for a variety of temporal and spatial scenarios. Time can be viewed as continuous or a discrete sequence of trials or time periods. Space can be viewed as continuous, or as a discrete grid of points at which events can occur. In two dimensions, results have been derived for square, rectangular, circular, triangular, and other shapes of scanning windows. The scan statistic S_w is the largest number of points in any window (for a fixed window size, shape, and orientation; or for a range of window sizes). The two-dimensional regions scanned include rectangles, the surface of a sphere, and more irregularly shaped geographical areas. In certain applications the events can only occur naturally at a discrete set of points in space. In other applications the underlying events can occur anywhere in space, but the method of observation limits events to a grid or lattice of points. Scan distributions have been derived for uniform, Poisson, and other distributed points in continuous space, and binomial, hypergeometric, and other distributions on two-dimensional grids. In the next section we discuss temporal scenarios.

43.2 Temporal Scenarios

One aspect of the scenario is whether its view is retrospective or prospective. A researcher might be reviewing events over some past time period of length T. The events might be a call for service, a reported cancer case, or an unacceptable item from an assembly line. In the *retrospective* case, the total number of events in the review period is a known number, N. The retrospective scan statistic analysis will typically be conditioned on N, a fixed known number, and in this case is referred to as either the *retrospective* or *conditional* case.

In other applications, the scientist uses scan statistics *prospectively* either to design a system's capacity to handle clustered demands, or to set up a monitoring system that will sound an alarm when an unusual cluster occurs. System capacity can be designed to give a specified small probability of overload within some future

period of operations of length T. Scan monitoring systems can be similarly designed so that (provided the process is "in control") there is, for example, only a 1% chance of a false alarm within a year; this is equivalent to saying that there is a 99% chance that the waiting time until a false alarm is greater than a year. Note that the total number of events, N, in time T is not known at either the system design time, or at the time an alarm is to be sounded. In the prospective case, the distribution of scan statistics cannot be conditioned on N. However, we often have information on the expected number of events in $(0, T)$. The prospective case is referred to as either the *prospective* or the *unconditional* case.

For each of the retrospective and prospective views, scan statistic distributions have been developed for continuous and for several discrete time scenarios. For example, the starting time of a hospital emergency room admission might be recorded to the nearest minute, and whether the patient had a particular syndrome may be recorded for each admission. For the event "admission of patient with syndrome," the scan statistic might be based on reported admission times, where time is viewed as a continuum. The *continuous scan statistic* is the maximum number of events in a window of length w that scans the time period $(0, T)$. In the continuous scenario, the times of occurrence of events are reported and for each time t in the review period, $(w \le t \le T)$, one knows the observed number of events $Y_w(t)$ and the expected number of events $E_w(t)$ in the subinterval $[t-w, t)$.

Alternatively, the analyst may only have a sequential list of patients available, and may only know whether or not each has the syndrome. In this case, the data is in the form of a discrete sequence of binary trials, and a *discrete* case scan statistic will be used. The data is viewed as a sequence of T trials, where for each trial whether or not an event has occurred is recorded; the discrete scan statistic is the maximum number of events in any w consecutive trials. For $t = w, w+1, \ldots, T$, $Y_w(t)$ and $E_w(t)$ are the observed and expected number of events within the w consecutive trials, $t-w+1, t-w+2, \ldots, t$.

In other cases, the reported data may only give hourly summary counts of patients with the syndrome, and the researcher may be keeping a moving sum of the number of such patients in the past six hours. In this discrete case we might use the *ratchet scan statistic*. In the ratchet scenario, time is divided into T disjoint intervals (hours, days, or weeks) and the reported data consists of the number of events in each interval. For $t = w, w+1, \ldots, T$, $Y_w(t)$ and $E_w(t)$ are the observed and expected number of events within the w consecutive intervals, $t-w+1, t-w+2, \ldots, t$.

Let S_w denote the scan statistic, $\max_t[Y_w(t)]$. For several important models for the above scenarios, exact formulae, approximations, and bounds are available for the distribution of S_w and related statistics. The following sections detail some of the most useful formula for the case where $E_w(t) = E_w$, a constant.

43.2.1 The Continuous Retrospective Case

The completely at random (constant background) model for this case is where N points (number of events) are independent uniform random variables over $(0, T)$.

$P(S_w \ge k | E_w, T)$ only depends on k, $N = (T/w)E_w$, and the ratio w/T. Choosing the units of measurement to make $T = 1$ simplifies the notation. A related scan statistic is the *minimum $(k-1)$th order gap*, W_k, the length of the smallest subinterval that contains k points. W_{k+1} is also referred to as the smallest *k-th-nearest neighbor distance* among the N points. The statistics are related by

$$P(S_w \ge k) = P(W_k \le w) \tag{43.1}$$

Now denote the common probability $P(k; N, w)$. W_N is the sample range, and W_2 is the smallest gap between any pair of points.

$$\begin{aligned} P(2; N, w) &= 1 - [1-(N-1)w]^N \,; \\ &\quad \text{for} \quad 0 \le w \le 1/(N-1) \\ &= 1 \,; \\ &\quad \text{for} \quad 1/(N-1) \le w \le 1 \,. \end{aligned} \tag{43.2}$$

$$\begin{aligned} P(N; N, w) &= Nw^{N-1} - (N-1)w^N \,. \\ &\quad \text{for} \quad 0 \le w \le 1 \,. \end{aligned} \tag{43.3}$$

For a given k and N, the expressions for $P(k; N, w)$ are piecewise polynomials in w with different polynomials for different ranges of w. A direct integration approach can be used to derive the piecewise polynomials for a few simple cases, but it becomes overly complex in general. An alternative combinatorial approach is used by [43.4] to derive the piecewise polynomials for $k > N/2$, with one polynomial for $w \le 0.5$, and another polynomial for $w > 0.5$. For $k > N/2$, $0 \le w \le 0.5$, the formula is particularly simple,

$$\begin{aligned} P(k; N, w) &= [(k - E_w)(1/w) + 1] \\ &\quad \times P(Y_t = k) + 2P(Y_t > k) \,, \\ &= \left[\left(kw^{-1} - \mathrm{N} + 1\right)\right. \\ &\quad \left. \times P(Y_t = k)\right] + 2P(Y_t \ge k) \,, \end{aligned} \tag{43.4}$$

where E_w is $N(w/T)$ and $P(Y_t = k)$ is the binomial probability $b(k; N, w)$

$$b(k; N, w) = \binom{N}{k} w^k (1-w)^{N-k} . \quad (43.5)$$

For $w > 0.5, k > (N+1)/2$, there are some additional terms involving binomials and cumulative binomial terms. This leaves the case $k \le N/2$ when $w \le 0.5$. Below we discuss exact results, tabled values, approximations and bounds that can be used to compute values for some cases. However, for hypothesis testing, [43.5] uses (43.4) as an accurate approximation for small to moderate (< 0.10, and even larger) $P(k; N, w)$ when $k \le N/2, w \le 0.5$.

In Example 1, there were 60 dates of death over a five-year period, and a one-year period with 23 or more deaths. Assuming that the 60 deaths were randomly distributed over the five-year period, $P(S_1 \ge 23 | T = 5) = P(23; 60, 0.2) = 0.029$, obtained by approximating using (43.4).

For certain applications one may seek to evaluate large values of $P(k; N, w)$, where the Wallenstein–Neff approximation may not be sufficiently accurate. In [43.3], Chapt. 8 discusses exact formulae, Chapt. 9 bounds, Chapt. 10 approximations, and Chapt. 2 details of the application of $P(k; N, w)$ to the continuous conditional (retrospective) case. We now give an overview of the types of exact results, other more accurate approximations, and bounds for $P(k; N, w)$.

Exact Results for $P(k; N, w)$

A general expression for $P(k; N, 1/L)$ (where L is an integer), in terms of sums of $L \times L$ determinants, is derived in [43.6]; This is generalized in [43.7] for $P(k; N, r/L)$ in terms of sums of products of several determinants, and simplified further by [43.8] in terms of sums of products of two determinants. These general formulae are computationally intense for small w small as they involve summing determinants of large matrices; however, these formulae can be used to generate the piecewise polynomials that can be used to compute the probabilities for any w. A procedure to do this is given and implemented by [43.9], for $N/3 < k, N/2$. A systematic approach to generating the polynomials is described in [43.10], Table 3, which lists the piecewise polynomials for $N \le 20$. The polynomials are then used to generate (in their Table 1a), $P(k; N, w)$ for $w \le 0.5, k \le N/2$, for $N \le 25$, with w to three decimal places. (Table 1 in [43.10] gives values for all k, for $N \le 25$, with w to two places.)

A powerful general *spacings* approach is derived in [43.11], and is used to find the distribution of W_k, the minimum of the sum of $k-1$ adjacent spacings between times of events. Huffer and Lin also find the maximum sum of $k-1$ adjacent spacings, which is related to the minimum number of events in a scanning window. They use their method to increase the range of values of N and k for which polynomials can be computed, with N as large as 61 for k close to $N/2$.

Approximations

A variety of approximations for $P(k; N, w)$ have been developed based on various combinations of approaches: methods with moments based on spacings, Poisson-type approximations with and without declumping, averaging bounds, using product limit approximations ([43.3], Chapt. 10). To emphasize the connection between higher order spacings and the scan statistic, we describe approximations based on using the method of moments applied to k-th order spacings or gaps. If $X_1 X_2 \ldots X_N$ are the ordered values of the N points in $(0, T)$, then $X_2 - X_1, X_3 - X_2, \ldots$ are the first-order spacings; $X_3 - X_1, X_4 - X_2, \ldots$ are the second-order spacings; and $X_k - X_1, X_{k+1} - X_2, \ldots, X_N - X_{N-k+1}$ are the $(k-1)$-order spacings. (Instead of spacings they are sometimes referred to as gaps, or quasi-ranges). Let $Z_k(w)$ denote the number of $(k-1)$-order spacings that are $\le w$.

$$P(S_w \ge k) = P(Z_k(w) \ge 1) = 1 - P(Z_k(w) = 0) . \quad (43.6)$$

The distribution of $Z_k(w)$ is complex, but it is straightforward to compute the expectation of $Z_k(w)$, and with more effort its variance.

$$\begin{aligned} E[Z_k(w)] &= (N-k+1) P(X_k - X_1 \le w) \\ &= (N-k+1) P(Y_t \ge k-1) . \end{aligned} \quad (43.7)$$

Here Y_t has the binomial distribution described in (43.5). In the method of moments we approximate the distribution of $Z_k(w)$ by a simpler distribution with some of the same moments. For example, choosing the approximating distribution to be Poissonian, with the same first moment, gives the approximation

$$P(S_w \ge k) \approx 1 - \exp[-(N-k+1) P(Y_t \ge k-1)] . \quad (43.8)$$

Note that the same Poisson model could be used to find $P(Z_k(w) \ge n)$, which could be used to approximate the distribution of the number of k-within-w clusters. Approximation (43.8) is not very good in general, because the $(N-k+1)$ overlapping $(k-1)^{st}$-order spacings are not independent. If $X_k - X_1$ is very small, this implies that $X_k - X_2$ is even smaller, which makes for

a greater chance that $X_{k+1} - X_2$ will also be small. A local declumping technique is used [43.12] to adjust for this type of association, and find an approximation of the form $1 - e^{-\mu}$. Approximations of this form, but with different μ values, have been used by [43.13] and others, and [43.14] proves limiting results of this form which suggest their use as approximations; however, care must be taken because the limiting results converge very slowly. *Glaz* in [43.15, 16] and other papers develops better approximations and a variety of bounds.

Moments for Continuous Retrospective Case

To compute the expectation, variance and other moments of S_w or W_k, one could average over the distribution of the statistic, where the cumulative distribution function of W_k is given by $P(k; N, w)$, and of S_w by $1 - P*(k+1; g, w)$. Using this method, [43.6] proves for $(N+1)/2 < k \le N$ that

$$\begin{aligned} E(W_k) &= [k - 2(N-k+1)b]/(N+1)\,, \\ \text{var}\,(W_k) &= (N-k+1)\big[(N+k+1) \\ &\quad + 2(2k-N-1)b \\ &\quad - 4(N+2)(N-k+1)b^2\big] \\ &\quad /(N+1)^2(N+2)\,, \end{aligned} \tag{43.9}$$

where b denotes the binomial term $b[N-k+1; 2(N-k+1), .5]$. Tables for the expectation and variance of W_k, for $\{k = 3, 4, 5; N = k(1)19\}$, $\{k = 6; N = 6(1)17\}$, $\{k = 7; N = 7(1)20\}$, $\{k = 8; N = 8(1)23\}$, and $\{k = 9; N = 9(1)25\}$ are generated in [43.10].

Averaging over exact and simulated values, [43.6] gives means and variances of S_w for $N \le 10$, $w = .1(.1).9$, and [43.17] tabulate means and variances of S_w for $N = 2(1)40, 40(5)70, 85, 100, 125, 150, 200(100)500, 1000$; and $w = 1/T$, $T = 3(1)6, 8, 12$.

43.2.2 Prospective Continuous Case

In certain applications, the researcher is interested in the distribution of the scan statistic given that the total number of events in $(0, T)$ is a random variable. Events are viewed as occurring at random times according to some process. The Poisson process is one completely-at-random chance model. In this process, the number of events $Y_w(t)$ in any interval $[t-w, t)$ is Poisson-distributed with mean E_w. $P[Y_w(t) = k] = p(k; E_w)$ for $k = 0, 1, 2, \ldots$, where $p(k; \lambda)$ denotes the Poisson probability $\exp(-\lambda)\lambda^k/k!$. For the Poisson process, $E_w = wE_1$ where E_1 is sometimes denoted λ; the numbers of events in any disjoint (not overlapping) intervals are independently distributed. There are various other ways to characterize the Poisson process. For the Poisson process, the arrival times between points are independent exponential random variables. Conditional on there being a total of N points from the Poisson process in $[0, T)$, these N points are uniformly distributed over $[0, T)$.

Given that events occur at random over time, let $T_{k,w}$ denote the waiting time until we first observe at least k events in an interval of length w. Formally, $T_{k,w}$ equals $X_{(i+k-1)}$ for the smallest i such that $X_{(i+k-1)} - X_{(i)} \le w$. The three scan statistics S_w, W_k, and $T_{k,w}$ are related by $P(S_w \ge k) = P(W_k \le w) = P(T_{k,w} \le T)$. These probabilities only depend on k, E_1 (the expected number of points in a window of length 1), and the ratio w/T. Denote the common probabilities for the Poisson model case by $P*(k; E_T, w/T)$, where $E_T = TE_1$. In computing $P(S_w \ge k)$ or $P(W_k \le w)$, the formula is sometimes simplified by choosing the scale of measurement to make $T = 1$ and by denoting E_1 by λ; when applying the simplified formula or using tabled values, care must be taken to interpret a λ consistent with the scale of measurement. To avoid confusion in what follows, we use the notation $P^*(k; E_T, w/T)$.

Exact and Approximate Formulae for Cluster Probabilities

In [43.18], asymptotic formulae are derived for $P^*(k; E_T, w/T)$, but these converge very slowly. The exact formulae in [43.8] for $P^*(k; E_T, w/T)$ are computationally intensive.

Table 2 in [43.10] gives $P^*(k; E_T, w/T)$ for $k = 3(1)9$, and a range of values for E_T and w/T. In Table 2, λ denotes E_T. Table 2a in [43.10] gives $P^*(k; 2E_w, 1/2)$, $P^*(k; 3E_w, 1/3)$, $P^*(k; 4E_w, 1/4)$ for $k = 3(1)9$, and a range of values for λ which denote $2E_w, 3E_w and 4E_w$ respectively in that table. Application (d) in [43.10, p. 4] illustrates how to use these values to accurately approximate $P^*(k; 2LE_w, 1/2L)$.

Reference [43.19] derives readily computable formulae for $P^*(k; 2E_w, 1/2)$ and $P^*(k; 3E_w, 1/3)$ and uses them to give the following highly accurate approximation for $P^*(k; E_T, w/T)$. Denote $1 - P^*(k; E_T, w/T)$, by $Q^*(k; E_T, w/T)$; $\exp(-\psi)\psi^j/j!$ by $p(j; \psi)$, and $\Sigma_{i \le k} p(j; \psi)$ by $F_p(k; \psi)$.

$$\begin{aligned} Q^*(k; E_T, w/T) \approx\; & Q^*(k; 2E_w, 1/2) \\ & \times [Q^*(k; 3E_w, 1/3) \\ & / Q^*(k; 2E_w, 1/2)]^{(T/w)-2}\,. \end{aligned}$$

$$
\begin{aligned}
Q^*(k;2\psi,1/2) &= [F_p(k-1;\psi)]^2 - (k-1) \\
&\quad \times p(k;\psi)p(k-2;\psi) \\
&\quad -(k-1-\psi)p(k;\psi) \\
&\quad \times F_p(k-3;\psi)\,, \\
Q^*(k;3\psi,1/3) &= (F_p(k-1;\psi))^3 \\
&\quad - A_1 + A_2 + A_3 - A_4\,, \qquad (43.10)
\end{aligned}
$$

where

$$
\begin{aligned}
A_1 &= 2p(k;\psi)F_p(k-1;\psi)[(k-1)F_p(k-2;\psi) \\
&\quad -\psi F_p(k-3;\psi)]\,; \\
A_2 &= 0.5[p(k;\psi)]^2\Big[(k-1)(k-2)F_p(k-3;\psi) \\
&\quad -2(k-2)\psi F_p(k-4;\psi)+\psi^2 F_p(k-5;\psi)\Big]\,, \\
A_3 &= \sum_{r=1}^{k-1} p(2k-r;\psi)[F_p(r-1;\psi)]^2\,, \\
A_4 &= \sum_{r=2}^{k-1} p(2k-r;\psi)p(r;\psi)[(r-1) \\
&\quad \times F_p(r-2;\psi)-\psi F_p(r-3;\psi)]\,.
\end{aligned}
$$

Subsequent researchers [43.20] note the remarkable accuracy of this approximation. Tight bounds for $Q^*(k;E_T,w/T)$ are derived by [43.21], who proves that approximation (43.10) falls within the bounds. Our experience is that it gives great accuracy over the entire range of the distribution. For example, $P^*(4,10;0.1) = 0.374$, $P^*(5;12,0.25) = 0.765$; $P^*(5;8,1/6) = 0.896$ by both the approximation and the exact tabled values in [43.10]. One can readily compute $Q^*(k;2E_w,1/2)$ and $Q^*(k;3E_w,1/3)$ for any k or E_w, or use tabled values. An even better approximation can be obtained by taking

$$
\begin{aligned}
Q^*(k;\lambda T,w/T) &\approx Q^*(k;\lambda,1/3) \\
&\quad \times [Q^*(k;4\lambda,1/4) \\
&\quad /Q^*(k;\lambda,1/3)]^{(T/w)-3}\,. \qquad (43.11)
\end{aligned}
$$

One can use values from Table 2a in *Neff* and *Naus* for $Q^*(k;\lambda,1/3)$ and $Q^*(k;4\lambda,1/4)$; or alternatively compute $Q^*(k;4\lambda,1/4)$ using the results of [43.8]. This generalizes naturally to even more accurate approximations. Recently [43.22] other highly accurate approximations for $Q^*(k;L\lambda,1/L) = Q_L$ have been developed, together with error bounds. Using terms of the form $Q^*(k;\lambda,1/r) = Q_r$, for $r = 2,3,4,\ldots$; for example, approximation (1.18) from that work uses Q_2 and Q_3, and has a relative error $< 3.3(L-1)(1-Q_2)^2$, for $L > 4$, if the error bound is small relative to 1. Approximation (1.17) uses Q_r for $r = 2,3,4,5$, and has a smaller error bound, under certain conditions.

A simpler approximation is derived by [43.23], which is computable on a calculator, and is reasonably accurate for small to moderate values of $P^*(k;\lambda T,w/T)$ that might be used when testing hypotheses for unusual clusters.

$$
\begin{aligned}
P^*(k;E_T,w/T) &\approx 1 - F_p(k-1;E_w) \\
&\quad \times \exp[-[(k-E_w)/k)] \\
&\quad \times \lambda(T-w)p(k-1;E_w)]\,. \qquad (43.12)
\end{aligned}
$$

For larger values of $P^*(k;E_T,w/T)$, (43.12) may not be accurate. For example, (43.12) gives $P^*(5,12,25) \approx 0.555$, as compared to the exact value of 0.765. In certain applications one seeks the distribution or moments of distribution of S_w, or the related statistics W_k, or $T_{k,w}$. If formulae for the moments are not available, one could average over the approximate distribution of the statistic, but in this case one would want to use an approximation that is accurate over the range of the distribution.

Example 2: A telecommunications engineer seeks to develop a system with the capacity to handle the possibility of multiple calls being dialed simultaneously. Dialing times start at random according to a Poisson process, with a 10 s dialing time. During an average 8 h busy period, 57 600 calls are dialed. The engineer asks how likely it is that at some point in the 8 h busy period there will be 50 or more phone calls being dialed simultaneously. There are an infinite number of overlapping intervals, each of 10 s duration, in an 8 h period. The maximum number of calls in any of the infinite number of overlapping windows is the scan statistic S_w. Here we are scanning a $T = 28\,800$ s period that has an expected number of calls $E_{28800} = 57600$, with a scanning window of $w = 10$ s, and asking how likely it is that $S_{10} \geq 50$. The answer needs to take into account the multiple comparisons involved in scanning the infinite number of overlapping 10 s periods within an 8 h period, and is given by $P^*(50;57600,10/28800)$, computed by (43.10) or (43.12).

Moments of Scan Statistic Distributions: Continuous Prospective Case

To compute the expectation, variance and other moments of S_w, W_k or $T_{k,w}$, one could average over the distribution of the statistic, where the cumulative distribution functions of W_k or $T_{k,w}$ are given by $P^*(k;E_T,w/T)$, and of S_w by $1 - P^*(k+1;E_T,w/T)$. To derive formula or compute the moments, one could use either the

exact formula or approximation (43.10), which is highly accurate over the range of the distribution.

Example 3: A window information flow control scheme is described in [43.24] where a sender of information packets stops sending when there is evidence of overload. An open-loop control mechanism avoids feedback delays by basing the control mechanism on the maximum number of information packets in a sliding time window of fixed prespecified length, the scan statistic. Strong approximations are used in [43.24] to derive asymptotic (for $T \gg w > \log T$) results for $P(S_w \geq k)$. For $w = 20$, $T = 1\,000\,000$ and a Poisson process with an average of one observation per unit of time, their asymptotic approximation gives $\mathrm{AVE}(S_w) = 42$, compared to their simulated value of 44.5. Reference [43.3] (pp. 33–34) uses approximation (43.10) to compute $Q^*(k; 1\,000\,000, 20/1\,000\,000)$ for $k = 41(1)52$), which gives all of the distribution needed to compute $AVE(S_w) = \Sigma_{k>1}[1 - Q^*(k)] \approx 44.84$. This is because for $k < 41$, $Q^*(k) < Q^*(41) \approx 6.6E-7$; and for $k > 52$, $Q^*(k) > Q^*(52) \approx 0.9999$.

Various approximations are given by [43.25] and [43.26] for moments of $T_{k,w}$. For the Poisson process, [43.26] gives approximations and bounds for the expectation and variance of $T_{k,w}$, and bounds for the expectation for general point processes with i.i.d. interarrival times between the points. Details are given for the Poisson, Bernoulli, and compound Poisson processes. We now discuss Samuel–Cahn's results for the Poisson case. Let $\delta_{k,w}$ denote the total number of points observed until the first cluster of k points within an interval of length w occurs. Note that $\delta_{k,w}$, $T_{k,w}$, S_w, and W_k are different but interrelated statistics associated with the scanning process. For a Poisson process with mean λ per unit time, the expected waiting times between points is $1/\lambda$. She applies Wald's lemma, to find for the Poisson case,

$$E(T_{k,w}) = E(\delta_{k,w})/\lambda \,, \tag{43.13}$$

and derives a series of approximations for $E(\delta_{k,w})$. The simplest of these is

$$E(\delta_{k,w}) \approx k + \left\{ [F_p(k-2; \lambda w)]^2 / P(\delta_{k,w} = k+1) \right\} , \tag{43.14}$$

where

$$P(\delta_{k,w} = k+1) = \sum_{i=0}^{k-2} (-1)^{k-2-i} p(i; \lambda w) + (-1)^{k-1} \exp(-2\lambda w) \,, \tag{43.15}$$

and where $p(i; \lambda w)$ and $F_p(k-2; \lambda w)$ are Poisson terms defined before (43.10).

43.2.3 Discrete Binary Trials: The Prospective Case

In start-up tests for a piece of equipment, the equipment might perform successfully on the first test trial, then fail on the second. Consecutive points in a QC chart may be in or out of a warning zone. In a stream of items sampled from an assembly line, some are defective while some are acceptable. Here the data is viewed as a sequence of T binary outcome trials. Each trial t results in a "success" or "failure." For $t = w, w+1, \ldots, T$, $Y_t(w)$ and $E_t(w)$ are the observed and expected number of "successes" within the w consecutive trials, $t-w+1, t-w+2, \ldots, t$.

The scan statistic S_w is the maximum number of successes within any w contiguous trials within the T trials. For the special case where $S_w = w$, a *success-run* of length w has occurred within the T trials. When $S_w = k$, a *quota* of k successes within m consecutive trials has occurred. Related statistics include W_k, the smallest number of consecutive trials that contain k ones; $T_{k,w}$, the number of trials until we first observe at least k ones in an interval of length w; and V_r, the length of the longest number of consecutive trials that have at most r failures. V_0 is the length of the longest success run. The statistics are related by $P(S_w \geq k) = P(W_k \leq w) = P(T_{k,w} \leq T)$, and $P(V_r \geq k+r) = P(S_{k+r} \geq k)$. We illustrate these statistics in the following example.

Example 4: The DNA molecule most often consists of two complementary strands of nucleotides each consisting of a deoxyribose residue, a phosphate group, and a nucleotide base. The four nucleotide bases are denoted A, C, G, T, where an A on one strand links with a T on the other strand, and similarly C with G. Molecular biologists sometimes compare DNA from two different sources by taking one strand from each, viewing each as a linear sequence of the letters A, C, G, T, aligning the two sequences by a global criterion, and then looking for long perfectly or almost perfectly matching "words" (subsequences). For illustration, consider the following two aligned sequences from two different plant proteins. If letters in the same position in the two sequences match, we put an "s" at that position; if not an "f"

Source 1: A A A C C G G G C A C T A C G G T G A G A C G T G A

Source 2: A A T C C C C C G T G C C C T T A G A G G C G T G G
Match: s s f s s f f f f f f f f s s s f s s s f s s s s f
The longest perfectly *matching word* is CGTG, corresponding to a success run of length 4. For the s/f sequence, $V_o = 4$. We note that the longest word with at most one mismatch is of length eight letters (underlined). Here $V_1 = 8$. If we had scanned the sequence of $T = 26$ aligned pairs of letters, looking for the largest number of matches within $w = 8$ letters, we would find $S_8 = 7$. If we had looked for the smallest number of consecutive letters containing seven matches, we would find $W_7 = 8$. The waiting time until we first observe seven matches within eight consecutive letters is $T_{7,8} = 25$.

Exact Results

There are a variety of algorithms to compute the distribution of the prospective discrete scan statistic exactly. Recursion relations and generating functions for the special cases of $k = w - 1$ and $k = w$ are given in [43.27]. The Markov chain imbedding approach was applied in [43.28] and [43.29] to derive an exact formula for the expected waiting time until a k-in-w quota, and is refined and unified in [43.30] to be efficient for computing the distributions of runs and scans. Useful recurrence relations and other formulae are given in [43.31–33]. Recently, a martingale approach has been used to find generating functions and moments of scan statistics [43.34].

Recent studies [43.35] and [43.36] into the computational complexity of the Markov chain approach to find exact results for the discrete scan statistic shows that it is computationally feasible. Many of the results are motivated by quality control and acceptance sampling applications [43.27, 33, 37]. Other recent results are motivated by the reliability of linear systems, where the system fails if any k within w consecutive components fail [43.1, 30, 38–40].

Approximate Results

There are asymptotic results for $P(S_w \geq k)$ for a variety of probability models that are called *Erdös–Rényi laws*, or Erdös–Rényi–Shepp theorems. DNA and protein sequence matching has stimulated further generalizations (see [43.41–44]). A simple random model is the Bernoulli trials model, where the T trials are independent binary trials, with a probability of "success" on trial t equal to a constant value p. Reference [43.45] reviews and proves some important general limit law results (as T tends to infinity), and shows how they apply in the special case of a Bernoulli process. The asymptotic results converge quite slowly, and for certain applications give only rough approximations ([43.3], pp. 233-235). Various approximations to $P(S_w k)$ are derived using the method of moments, a Poisson approximation using declumping, and other methods [43.46].

For the Bernoulli process, denote $P(S_w \geq k)$ by $P'(k|w; T; p) = 1 - Q'(k|w; T; p)$.

In [43.19], the following highly accurate approximation is given for $Q'(k|w; T; p)$. Let $b(k; w, p)$ be the binomial probability defined in (43.5), and let

$$
\begin{aligned}
F_b(r; w, p) &= \Sigma_{i<r} b(i; w, p)\,, \\
Q'(k|w; T; p) &\approx Q'(k|w; 2w; p)\left[Q'(k|w; 3w; p)\right. \\
&\quad \left./\,Q'(k|w; 2w; p)\right]^{(T/w)-2}\,, \\
Q'(k|w; 2w; p) &= [F_b(k-1; w, p)]^2 \\
&\quad -(k-1)b(k; w, p) \\
&\quad \times F_b(k-2; w, p) \\
&\quad + wpb(k; w, p) \\
&\quad \times F_b(k-3; w-1, p)\,, \\
Q'(k|w; 3w; p) &= (F_b(k-1; w, p))^3 \\
&\quad - A_1 + A_2 + A_3 - A_4\,, \qquad (43.16)
\end{aligned}
$$

where

$$
\begin{aligned}
A_1 &= 2b(k; w, p)F_b(k-1; w, p)\,[(k-1) \\
&\quad \times F_b(k-2; w, p) - wpF_b(k-3; w-1, p)]\,; \\
A_2 &= 0.5[b(k; w, p)]^2\left[(k-1)(k-2)\right. \\
&\quad \times F_b(k-3; w, p) \\
&\quad -2(k-2)wpF_b(k-4; w-1, p) \\
&\quad \left.+(w-1)p^2F_b(k-5; w-2, p)\right]\,; \\
A_3 &= \sum_{r=1}^{k-1} b(2k-r; w, p)[F_b(r-1; w, p)]^2\,; \\
A_4 &= \sum_{r=2}^{k-1} b(2k-r; w, p)b(r; w, p)\,[(r-1) \\
&\quad \times F_b(r-2; w, p) - wpF_b(r-3; w-1, p)]\,.
\end{aligned}
$$

The following simpler, and fairly accurate, approximation is suggested in [43.47].

$$
P'(k|w; T; p) \approx 1 - \left[C(D/C)^{(T/w)-2}\right]\,, \qquad (43.17)
$$

where

$$
\begin{aligned}
C &= 2\Sigma_{i<k} b(i; w, p) - 1 - (k-1-wp)b(k; w, p)\,, \\
D &= 2\Sigma_{i<k} b(i; w, p) - 1 - (2k-1-2wp) \\
&\quad \times b(k; w, p)\,.
\end{aligned}
$$

Example: $k = 8$, $w = 10$, $p = 0.3$, $T = 50$. Applying (43.16), $Q'(8|10; 20; 0.3) = 0.991032$; $Q'(8|10; 30; 0.3) = 0.983827$; $P'(8|10; 50; 0.3) \approx 0.03$.

Applying (43.17) gives $C = 0.991032$; $D = 0.9837985$; $P'(8|10; 50; 0.3) \approx 0.03$. Both of these results agree with the exact result from the Markov Chain imbedding approach.

In some applications the researcher is interested in scanning for unusual clusters of successes or clusters of failures. One can use the above results to bound the probability of either type, or can use the Markov Chain imbedding or another approach to compute the probability directly [43.39]. In other applications, individual trials can result in an integer number of points and [43.48] derives accurate approximations and tight bounds for this case.

Moments of Discrete Trial Scan Statistics

An accurate approximation for the expected waiting time until a k in w cluster in a Bernoulli trials, with a probability p of *success* on an individual trial, is given in [43.19]

$$E(T_{k,w}) \approx 2w + Q'(k|w; 2w; p) \Big/ \left\{1 - [Q'(k|w; 3w; p) \Big/ Q'(k|w; 2w; p)]^{1/w}\right\} . \quad (43.18)$$

More complicated formulae for the exact expectation, and methods to find it, are given in [43.28, 29], and other approximations in [43.26] for the expectation and variance of the waiting time for the Bernoulli and more general processes.

An important generalization is to the case of independent binary trials with unequal probabilities. For many results see [43.1]. Reliability engineers have been studying the reliability of a linearly ordered set of N independent components with different probabilities of being defective. In the *k-within-consecutive-m-out-of-N systems*, the system fails if there is a quota of k defectives within any m consecutive components in the system (see [43.40, 49]).

43.2.4 Discrete Binary Trials: The Retrospective Case

In this discrete scenario, the data is viewed as a sequence of T binary outcome trials. Each trial t results in a "success" or "failure." S_w is the largest number of successes in any w consecutive trials. A simple random model is where there are N successes distributed at random over the T trials. Denote $P(S_w \geq k)$ by $P(k|w; N, T)$.

The exact distribution of $P(k|w; N, T)$ for the case $k > N/2$ is given in [43.31]. This formula can also be used to approximate small values of $P(k|w; N, T)$ for $k < N/2$, and the approximation is given below in (43.19). Reference [43.50] derives the exact distribution for all N and k, for $m/N = 1/L$, L an integer, and more generally for $m = cR$, $N = cL$, where c, R, and L are integers, and $c > 1$. These general formulae are computationally complex, but can be used to derive simpler formulae for special cases. These formulae can also be used to approximate other cases.

Approximations and bounds for $P(k|w; N, T)$ are given in [43.3] (pp. 56–58, 212–216) and in [43.1] (pp. 319–323). These can then be used to approximate the moments of the distributions of S_w and W_k. A simple approximation is given by

$$P(k|w; N, T) = [(k - E_w)(1/w) + 1]P(Y_t = k) + 2P(Y_t > k) , \quad (43.19)$$

where

$$E_w = N(w/T) ;$$

$$P(Y_t = k) = H(k, N, w, T) = \binom{w}{k}\binom{T-w}{N-k} \Big/ \binom{T}{N} .$$

For the case where w and T are large, the discrete retrospective scan probability $P(k|w; N, T)$, can be approximated by the continuous retrospective scan probability $P(k; N, w/T)$.

43.2.5 Ratchet-Scan: The Retrospective Case

In the ratchet scenario, time is divided into T disjoint intervals (hours, days, or weeks) and the reported data consists of the number of events in each interval. For $t = w, w+1, \ldots, T$, $Y_t(w)$ and $E_t(w)$ are the observed and expected number of events within the w consecutive intervals, $t - w + 1, t - w + 2, \ldots, t$. For the constant background retrospective case there are N events spread at random over the T intervals. (The model is multinomial where each of the N balls is equally likely to fall in any of the T cells, independently of the other balls.) Here $\lambda = wN/T$. Approximations and bounds for $P(k; \lambda, w, T)$ are described in [43.3] (pp. 327–328), [43.2] (p. 81–91), and [43.51]. The simple Bonferroni upper bound is

$$P(k; \lambda, w, T) < (T - w + 1)\Sigma_{i \geq k} b(i; N, p) , \quad (43.20)$$

where $N = T\lambda/w$; $p = 1/T$. Note also that $P(k; \lambda, w, T)$ for the ratchet scan must be less than that for the continuous scan. Thus, $P(k; \lambda, w, T)$ for the retrospective ratchet scan must be less than the right-hand side of (43.5)

43.2.6 Ratchet-Scan: The Prospective Case

In the ratchet scenario, time is divided into T disjoint intervals (hours, days, or weeks) and the reported data consists of the number of events in each interval. For $t = w, w+1, \ldots, T$, $Y_t(w)$ and $E_t(w)$ are the observed and expected number of events within the w consecutive intervals, $t-w+1, t-w+2, \ldots, t$. For the constant background prospective case the number of events (counts) in the T intervals are T independently and identically distributed Poisson random variables, each with expectation λ/w. For the prospective ratchet scan constant background case, the approach in [43.5] gives the following approximation:

$$P(k; \lambda, w, T) \approx (T-w+1)G(k, \lambda) - (T-w) \times \Sigma_{j=0} p(j, \lambda)[G(k-j, \lambda/w)]^2 , \quad (43.21)$$

Here $p(j, \lambda)$ is the Poisson probability defined in (43.20), and

$$G(k, \lambda) = \Sigma_{j \geq k} p(j, \lambda) .$$

For the case where the number of events in the T intervals are independently and identically distributed random variables, [43.48] derives accurate approximations and tight bounds for $P(k; \lambda, w, T)$.

43.2.7 Events Distributed on the Circle

When studying seasonality effects of disease patterns over time, one might be interested in unusual clustering during certain periods of the year. The researcher may view the time period as a circle, with January following December. In studying the directions of flights of birds or insects, the directions may be viewed as points on a circle. Many of the distributional results described above for the line have also been derived for scan statistics on the circle. The circular ratchet scan was introduced in [43.52] and further developed in [43.51]; accurate approximations for the continuous scan on the circle and line are given in [43.19]; and for the discrete scan on the circle in [43.53].

43.3 Higher Dimensional Scans

Scan statistic have been applied extensively to study the clustering of diseases over space. In two dimensions, circular, rectangular, elliptical and other shaped scanning windows are used. Regions scanned can have arbitrary shapes (typical in epidemiology), or the surface of a sphere (astronomy or ships at sea), or approximately rectangular (blood cells on a slide). In three dimensions, two of which may be spatial and one temporal, cylindrical scanning windows may be used. In the case where all three dimensions are spatial, a scanning sphere or cube may be used. Scan distributions have been approximated or simulated for events from homogeneous or heterogeneous Poisson processes, uniform, or more generally distributed points in continuous space, and for various models for two-dimensional grids. Simulation is the most widely applied approach, but can be computationally intensive for a large number of points. Not only must the points be generated, but one must check all possible positions of the scanning window, and in variable-size window applications [43.54–57] each of the window sizes must be checked too. A variety of algorithms have been developed to simulate scan probabilities in two dimensions. Efficient Monte Carlo algorithms for one and two dimensions are developed in [43.55, 58, 59]; (see [43.3], Chapt. 16). Importance sampling can be used to reduce the computational effort of the simulation [43.60]. A web-based program, SatScan [43.56], is available that scans generally distributed points over an arbitrarily shaped region with circular scanning windows (with a variety of diameters).

In two dimensions, the points are randomly distributed over a two-dimensional region. A circular (or rectangular or other shaped) window scans the region. The scan statistic S_w is the largest number of points in any window of diameter w; the scan statistic W_k generalizes to the diameter of the smallest scanning window that contains k points. The distributions of the statistics S_w and W_k are still related, $P(S_w \geq k) = P(W_k \leq w)$. However, in more than one dimension, W_{k+1} is not equivalent to the smallest k-th-nearest neighbor distance among the N points. Section 43.3.1 discusses the retrospective case of a fixed number of points in the unit square; Sect. 43.3.2 discusses the prospective case where the number of points is Poisson distributed.

43.3.1 Retrospective Continuous Two-Dimensional Scan

We first focus on the problem of scanning a rectangular region with a rectangular window with sides of length u and v that are oriented parallel to the sides of the square. There are a fixed number, N, of points distributed over the rectangular region, The units of measurement are chosen to make the rectangular region the unit square. The results for the unit square give the results for an $a \times b$ scanning window within an $S \times T$ rectangular region, by choosing the units of measurement for the $x-$ and $y-$axes to make $S = 1$ unit on the x-axis, and $T = 1$ unit on the y-axis, and by setting $u = a/S$ and $v = b/T$.

Given N points distributed at random over the unit square, let $S_{u,v}$ denote the maximum number of points in any subrectangle with sides of length u and height v parallel to the sides of the unit square. Let $P(k; N, u, v)$ denote $P(S_{u,v} \geq k)$. Bounds for $P(k; N, u, v)$, that converge for small u, v are given in [43.61], and the approximation for this case is

$$P(k; N, u, v) \cong k^2 \binom{N}{k} (uv)^{k-1} . \tag{43.22}$$

An exact formula is available for $P(N-1; N; u, v)$, (see [43.3], Chapt. 16). The following approximation by [43.62] is based on large deviation theory

$$\begin{aligned} P(k; N, u, v) \cong & \left\{ \left[N^2 w(1-u)(1-v)E^3 \right. \right. \\ & \left. \left. /(1-w)^3(1+E) \right] + C \right\} \\ & \times b(k; N, w) , \end{aligned} \tag{43.23}$$

where

$$\begin{aligned} w &= uv,\ E = (k/Nw) - 1 ; \\ b(k; N, w) &= \binom{N}{k} w^k (1-w)^{N-k} , \\ C &= [Nv(1-u)E/(1-w)] \\ &\quad + \left[Nu(1-v)E^2/(1+E)(1-w)^2 \right] \\ &\quad + [(1+E)(1-w)/E] . \end{aligned}$$

The approximation is refined further in [43.62], but (43.23) appears to give good accuracy for small $P(k; N, u, v)$. Simulation is used to evaluate larger values of $P(k; N, u, v)$. Order the points by the X coordinates, with $X_1 < X_2 \leq ... \leq X_N$. For each $i = 1, ..., N - k + 1$, check whether $(X_{k+i-1} - X_i \leq u)$, and if so, whether the corresponding Y's fall within a distance v.

For the case of a circular scanning window of radius r, the algorithm looks at pairs of points within a distance $2r$, finds the two circles of radius r on which the two points fall on the circumference, and counts the number of points in each of the circles.

The above generalizes to higher dimensions. Scan the r-dimensional unit cube with a rectangular block with sides of length $u_1, ..., u_r$, oriented parallel to the sides of the unit cube. Let $w = \prod u_i$ denote the volume of the scanning rectangle. Let $P(k; N, u_1, u_2, ..., u_r)$ denote the probability that at least one scanning rectangular block with sides $(u_1, u_2, ..., u_r)$ parallel to those of the unit cube contains at least k points. In corollary 2.3, [43.63] gives the approximation for $r = 1, 2, 3$

$$\begin{aligned} P(k; N, u_1, u_2, ..., u_r) \cong & \{1 - [Nw/k(1-w)]\}^{2r-1} \\ & \times (k^r/w) b(k : N, w) , \end{aligned} \tag{43.24}$$

where $w = \prod u_i$, and $b(k; N, w)$ is the binomial probability in (43.23). For the case $r = 2$, (43.24) does not reduce exactly to (43.23), but gives similar values for small probabilities.

43.3.2 Prospective Continuous Two-Dimensional Scan

Let the number of points in the unit square be a Poisson-distributed random variable with mean λ. Scan the unit square with a subrectangle with sides of length u and v that are parallel to the sides of the square. Let $P^*(k; \lambda, u, v)$ denote the probability that at least one uxv-scanning subrectangle contains at least k points. Several approximations from [43.64] and [43.65] are given for $P^*(k; \lambda, u, v)$, with the best approximation being (from [43.65]):

$$\begin{aligned} P^*(k; \lambda, u, v) \cong & 1 - F_p(k-1; \lambda uv) \\ & \exp\{-\zeta(1 - (\lambda uv/k))\lambda v(1-u) \\ & \times p(k-1; \lambda uv)\} , \\ \zeta = & [1 - (\lambda uv/k)]\lambda u(1-v) \\ & \times [P^*(k-1; \lambda v, u) \\ & - P^*(k; \lambda v, u)] ; \\ F_p(k-1; \lambda uv) = & \sum_{i=0}^{k-1} p(i; \lambda uv) , \end{aligned} \tag{43.25}$$

where $P^*(k; \lambda v, u)$ is the one-dimensional scan statistic. A simpler but rougher approximation is

$$\begin{aligned} P^*(k; \lambda, u, v) \cong & 1 - [1 - P^*(k-1; \lambda v, u)] \\ & \exp(-\zeta) . \end{aligned} \tag{43.26}$$

For the case of scanning an $S \times T$ rectangular region with a circular window of radius r, let $r\pi^{0.5}/S = u$, $r\pi^{0.5}/T = v$, and choose the X- and Y-scales so that $S = 1$ and $T = 1$. Let $P_c^*(k; \lambda, u, v)$ denote the probability that the maximum number of points in the scanning window is at least k, where λuv denotes the expected number of points in a circle of radius r. The following approximation is given in [43.64]:

$$P_c^* \cong 1 - \exp[-k\lambda(1-2u)(1-2v)p(k-1; \lambda uv)] . \quad (43.27)$$

The simulations in [43.65, 66] show that for small u, v, $P_c^* \cong P^*(k; \lambda, u, v)$. For small u, v, we approximate the circular window by a square window of the same area.

Researchers continue to develop more accurate approximations for the distribution of the two-dimensional scan statistic for the Poisson process; see [43.67] and [43.25]. Other researchers seek to generalize the asymptotic results and some approximations to higher dimensions or more general scanning regions. The approach of [43.65] is generalized in [43.66] to give approximations for more than two dimensions. Recent research [43.68] derives approximate results with error bounds for one and higher dimensions for general distributions, and general bounded scanning sets, and uses these results to prove various asymptotic results. Poisson approximation and large deviation theory has been used in [43.69] to derive very general results for scanning spatial regions of two and more dimensions.

43.3.3 Clustering on the Lattice

In some applications, events can only occur on a grid of points; a recent example [43.70] deals with the reliability of a two-dimensional system. Another example involves clustering of diseased plants in a field of evenly spaced plants. In other applications, the method of measurement limits the observed events to occurring on a rectangular R by T lattice of points. The researcher scans the grid looking for unusual clusters. Results have been developed for several models; The researcher scans the lattice with a rectangular m_1 by m_2 sublattice with sides parallel to those of the lattice; events are independent and equally likely to occur at any point of the lattice. This case is the discrete analog of the continuous prospective case with an oriented rectangular scanning window, as discussed in Sect. 43.3.2.

Let X_{ij} for $i = 1, .., R$; $j = 1, ..., T$ denote a rectangular lattice of independent and identically distributed Bernoulli random variables, where $P(X_{ij} = 1) = p = 1 - P(X_{ij} = 0)$. View the lattice with position (1,1) in the lower left corner. Let $Y_{r,s}(m_1, m_2)$ denote the number of events (ones) in an m_1 by m_2 subrectangle whose lower left corner is at (r, s) in the lattice. Denote the *two-dimensional discrete scan statistic* $S'_{m1,m2}$, the maximum number of events in the scanning subrectangle when we scan the R by T lattice with an m_1 by m_2 rectangle of points (with sides oriented with the sides of the lattice).

Algorithms are given in [43.71] to find the largest rectangle with all 1's in the lattice, and [43.71] generalizes the limit law to higher dimensions, and to allow for some 0's in the subrectangle. Approximations for $P\left(S'_{m1,m2} \geq k\right)$ are given in [43.72]. For the special case of a square lattice, $R = T$, and a square scanning subrectangle $m_1 = m_2 = m$, let B_s denote the event $[Y_{1,s}(m, m) < k]$. Then,

$$P\left(S'_{m,m} \geq k\right) \cong 1 - q(2m-1) \times [q(2m) / q(2m-1)]^{(T-2m+1)(T-m+1)} , \quad (43.28)$$

where

$$q(2m-1) = P(B_1 B_2 ... B_m) ,$$
$$q(2m) = P(B_1 B_2 ... B_{m+1}) .$$

The terms $q(2m-1)$ and $q(2m)$ can be evaluated using an algorithm in [43.73].

For m_1, m_2, R, and T all large, the discrete scan probability can be approximated by the continuous case probability in Sect. 43.3.2.

$$P^*(k; \lambda, u = m_1/R; v = m_2/T) \cong P\left(S'_{m,m} \geq k\right) . \quad (43.29)$$

43.4 Other Scan Statistics

43.4.1 Unusually Small Scans

Scan statistics have been developed to test for unusually sparse intervals of time, or regions of space. Given N points independently drawn from the uniform distribution on [0, 1), let D_w denote the smallest number of points in an interval of length w; let V_k denote the size of the largest subinterval of [0, 1) that contains k points.

The statistics are related:

$$P(D_w \le k) = P(V_k \ge w)\,. \tag{43.30}$$

The scan statistic V_{k+1} can be viewed as the maximum of the sum of k consecutive spacings. The interval V_{r+1} is called the "maximum r-th-order gap".

A variety of results have been derived for the related statistics D_w and V_k. In [43.14] asymptotic results are derived for the distributions of order statistics of r-th order gaps, for i.i.d. distributed spacings (gaps between consecutive points) and some more general uniform mixing stationary processes. For the special case of a minimum r-scans of $n(= N+1)$ i.i.d. uniform [0, 1) spacings, they give

$$\lim_{n->\infty} P\{V_{r+1} \le [\log_e n + (r-1) \times \log_e \log_e n + y]/n\} = \exp[\,e^{-y}/(r-1)!]\,. \tag{43.31}$$

The asymptotic convergence of (43.31) is very slow, and care must be taken in using it as an approximation. For the case of N uniformly distributed points on the line, the exact distribution of D_w and V_k are derived in [43.74]. The formulae are computationally intensive, but can be applied to derive special cases that can be used to find highly accurate approximations similar in form to (43.16). A general approach to finding the distribution of V_k is described in [43.11].

For a sequence of Bernoulli trials, the distribution of the minimum scan can be computed using the formula for the maximum scan. Let D_w^* denote the minimum number of successes in any w consecutive trials. Then $D_w^* \ge k$ if the maximum number of failures in any w trials is less than or equal to $w-k$.

The distribution of the minimum scan for the line and circle is related to a multiple coverage problem. The multiple coverage problem is as follows: Given N subarcs each of length w dropped at random on the circumference of the unit circle, what is the probability that the arcs completely cover the circumference of the circle at least m times? To relate the coverage to the scan problem, let the N points in the scan problem correspond to the midpoints of the subarcs in the coverage problem. If the N subarcs do not cover the circle m times, there must be some point on the circumference not covered enough. This implies that the subinterval of length w centered at that point contains fewer than m of the N midpoints of the N subarcs; in this case, the minimum number of the N (mid)points in a scanning window of length w must be less than m.

43.4.2 The Number of Scan Clusters

There are several ways to count clusters depending on the overlap allowed between two clusters. For example, suppose we scan (0,1) with a window of width $w = 0.20$, looking for clusters of at least $k = 3$ points. Assume that we observe the six points: 0.10, 0.15, 0.28, 0.34, 0.41, 0.62. One approach to counting the number of 3-within-0.20 clusters is to see how many of the 2-spacings are $< w$. Here, $0.28 - 0.10 = 0.18 < w$, $0.34 - 0.15 = 0.19 < w$, $0.41 - 0.28 = 0.13 \le w$, and $0.62 - 0.34 > w$. We would count three clusters. Note that two clusters share some (but not all) points, and this case is sometimes referred to as a nonoverlapping case. For the conditional case, expressions are available for the expectation and variance of the number of clusters. See [43.75] and [43.11] for this method of counting. Results are derived in [43.14] for the distribution of the number of clusters that do not overlap with a previously counted cluster A Markovian declumping is used in [43.76] and [43.77], where an r-spacing$< w$ is counted so long as the immediately previous r-spacing is not counted. A compound Poisson approach to counting the number of clumps is applied in [43.11, 76, 78] and [43.12]. This is generalized in [43.79] to two and higher dimensions. See ([43.3], Chapt. 17) for a summary of many of the results for the continuous scan.

For a sequence of trials, ([43.1], Chapt. 10) describes in detail the Markov chain approach to finding the distribution of the waiting time until the k-th discrete scan cluster, for different ways of counting clusters. In Chapt. 4, [43.1] discusses the discrete counting approach for the case of runs, and there is a great deal of literature on the number of overlapping and nonoverlapping runs; see also [43.80]. In [43.81], a compound Poisson approximation is used for the multiple cluster problem, and the motivating example is the clustering of individual claims exceeding threshold risks.

43.4.3 The Double-Scan Statistic

In [43.82], a scan-type statistic called the *double-scan statistic* is defined based on the number of "declumped" (a type of nonoverlapping) clusters that contain at least one of each of two types of event. The expectation and approximate distribution of the number of declumped clusters is derived for this test statistic for two chance models. Define the event $E(i)$ to have occurred if there are at least one of each of the two types of events anywhere within the w consecutive days $i, i+1, \ldots, i+w-1$. The event $E(i)$ indicates the occurrence

of a two-type w-day cluster. Let $Z(i) = 1$ if $E(i)$ occurs and none of $E(i-1)$, $E(i-2)$, ..., $E(i-w+1)$ occur; $Z(i) = 0$ otherwise. Let $S_{w(2)} = \Sigma_{1<i<N-w+1} Z_i . S_{w(2)}$ is the *double scan statistic*. This method counts the number of times that an $E(i)$ occurs with no previously overlapping $E(i)$'s. When the events are relatively rare and distributed according to certain chance models, the number of these declumped clusters is approximately Poisson-distributed [43.64]. This model fits well when the two types of events do not occur too frequently. As the density of the number of events increases, there comes a point where more events lead to fewer clusters. In [43.83], the distributions of a family of double-scan statistics are derived using a different approach to count clusters. This method treats clusters as recurrent events, and counts the number of times that an $E(i)$ occurs with no previously overlapping $E(i)$'s that were counted in a cluster.

The double-scan statistic can be generalized, for the case of two types of events, to clusters where there are at least r type one and s type two events within a w-day period. For other applications, the statistics can be generalized to more than two types of events, and the distribution of the number of declumped clusters can be derived.

43.4.4 Scanning Trees and Upper Level Scan Statistics

The scanning approach has been extended to a variety of data structures. In two-dimensional geographic scanning, the SatScan [43.56] simulation approach uses a range of circular window sizes to detect unusual clusters. When studying patterns of disease, a researcher may be looking for connected regions (such as counties) with above-average concentrations of disease. These regions may not all fall within a compact circular, elliptical or other simple shape. One can still scan for these linked high-density regions, and [43.84] develops a "higher-level scan" approach for assessing the unusualness of clusters that arise.

In other situations, the researcher may not be looking for clusters of disease in space, but instead for some other variable such as occupation. One might try to scan a data set combining occupations that have above-average incidence. However, scanning all possible combinations involves so many multiple comparisons that the test statistic adjusted for this would not be powerful, and the results would be difficult to interpret. In the case of occupations (and many other variables) the data can be placed in the form of a tree structure (for example, a carpenter, electrician, plumber are all under building trades). In [43.85] an approach is given to scanning data in a tree structure, looking for unusual clusters, and adjusting for the multiple comparisons made.

In some applications one seeks to scan with a range of window sizes. Simulation is typically used to test unusual clusters [43.54,56]. Asymptotic results [43.62,63, 68,69] and approximations [43.57] are also available for a variety of models.

References

43.1 N. Balakrishnan, M.V. Koutras: *Runs and Scans with Applications*, Vol. 1 (Wiley, New York 2001)

43.2 J. Glaz, N. Balakrishnan: *Scan Statistics and Applications* (Birkhauser, Boston 1999)

43.3 J. Glaz, J. Naus, S. Wallenstein: *Scan Statistics* (Springer, Berlin Heidelberg 2001)

43.4 J. I. Naus: The distribution of the size of the maximum cluster of points on a line, J. Am. Stat. Assoc. **60**, 532–538 (1965)

43.5 S. Wallenstein, N. Neff: An approximation for the distribution of the scan statistic, Stat. Med. **6**, 197–207 (1987)

43.6 J. I. Naus: Some probabilities, expectations, variances for the size of the largest clusters, smallest intervals, J. Am. Stat. Assoc. **61**, 1191–1199 (1966)

43.7 S. Wallenstein, J. Naus: Probabilities for a k–th nearest-neighbor problem on the line, Ann. Prob. **1**, 188–190 (1973)

43.8 R. J. Huntington, J. I. Naus: A simpler expression for K–th nearest neighbor coincidence probabilities, Ann. Prob. **3**, 894–896 (1975)

43.9 S. Wallenstein, J. Naus: Probabilities for the size of largest clusters, smallest intervals, J. Am. Stat. Assoc. **69**, 690–697 (1974)

43.10 N. Neff, J. Naus: *Selected Tables in Mathematical Statistics, Volume VI: The distribution of the Size of the Maximum Cluster of Points on a Line* (Am. Math. Soc., Providence 1980)

43.11 F. W. Huffer, C-T. Lin: Computing the exact distribution of the extremes of sums of consecutive spacings, Comput. Stat. Data Anal. **26**, 117–132 (1997)

43.12 J. Glaz, J. Naus, M. Roos, S. Wallenstein: Poisson approximations for the distribution, moments of ordered m-spacings, J. Appl. Prob. **31A**, 271–81 (1994)

43.13 N. Cressie: On the minimum of higher order gaps, Australian J. Stat. **19**, 132–143 (1977)

43.14 A. Dembo, S. Karlin: Poisson approximations for r-scan processes, Ann. Appl. Prob. **2**, 329–357 (1992)
43.15 J. Glaz: Approximations, bounds for the distribution of the scan statistic, J. Am. Stat. Assoc. **84**, 560–569 (1989)
43.16 J. Glaz: Approximations for tail probabilities, moments of the scan statistic, Stat. Med. **12**, 1845–1852 (1993)
43.17 S. Wallenstein, M. S. Gould, M. Kleinman: Use of the scan statistic to detect time-space clustering, Am. J. Epidemiology **130**, 1057–1064 (1989)
43.18 G. F. Newell: *Distribution for the Smallest Distance Between any Pair of K–th Nearest-Neighbor Random Points on a Line*, ed. by M. Rosenblatt (Wiley, New York 1963) pp. 89–103
43.19 J. I. Naus: Approximations for distributions of scan statistics, J. Am. Stat. Assoc. **77**, 177–183 (1982)
43.20 M. Berman, G. K. Eagleson: A useful upper bound for the tail probability of the scan statistic when the sample size is large, J. Am. Stat. Assoc. **84**, 560–566 (1985)
43.21 S. Janson: Bounds on the distributions of extremal values of a scanning process, Stochastic Proc. Appl **18**, 313–328 (1984)
43.22 G. Haiman: Estimating the distributions of scan statistics with high precision, Extremes **3**, 4349–361 (2000)
43.23 S. E. Alm: On the distribution of a scan statistic of a Poisson process. In: *Probability and Mathematical Statistics*, ed. by A. Gut, L. Helst (Univ. Press, Upsalla 1983) pp. 1–10
43.24 A. W. Berger, W. Whitt: Asymptotics for open-loop window flow control, J. App. Math. Stochastic Anal. , 70 (1993) Special issue in honor of Lajos Takac's 70^{th} birthday
43.25 M. Mansson: Poisson approximation in connection with clustering of random points, Ann. Appl. Prob. **9**, 465–492 (1999)
43.26 E. Samuel-Cahn: Simple approximations to the expected waiting time for a cluster of any given size, for point processes, Adv. Appl. Prob. **15**, 21–38 (1983)
43.27 S. W. Roberts: Properties of control chart zone tests, Bell System Tech. J. **37**, 83–114 (1958)
43.28 I. Greenberg: The first occurrence of N successes in M trials, Technometrics **12**, 627–634 (1970)
43.29 R. J. Huntington: Mean recurrence times for k successes within m trials, J. Appl. Prob. **13**, 604–607 (1976)
43.30 M. V. Koutras, V. A. Alexandrou: Runs, scans, urn model distributions: A unified Markov chain approach, Ann. Inst. Stat. Math. **47**, 743–766 (1995)
43.31 B. Saperstein: The generalized birthday problem, J. Am. Stat. Assoc. **67**, 425–428 (1972)
43.32 B. Saperstein: Note on a clustering problem, J. Applied Prob. **12**, 629–632 (1975)
43.33 B. Saperstein: The analysis of attribute moving averages: MIL-STD-105D reduced inspection plans, Sixth Conf. Stochastic Proc. Appl. Tel Aviv (1976)
43.34 V. J. Pozdnyakov, J. Glaz, M. Kulldorff, M. Steele: A martingale approach to scan statistics, Ann. Inst. Stat. Math. (2004) in press
43.35 F. K. Hwang, P. E. Wright: An O(n/log n) algorithm for the generalized birthday problem, Comp. Stat. Data Anal. **23**, 443–451 (1997)
43.36 H. P. Chan, T. L. Lai: Boundary crossing probabilities for scan statistics, their applications to change point detection, Method. Comp. Appl. Prob. **4**, 317–336 (2002)
43.37 G. Shmueli: System wide probabilities for systems with runs, scans rules, Method. Comp. Appl. Prob. **4**, 401–419 (2003)
43.38 J. C. Chang, R. J. Chen, F. K. Hwang: A minimal-automation-based algorithm for the reliability of Con(d,k,n) system, Method. Comp. Appl. Prob. **3**, 379–386 (2001)
43.39 J. C. Fu: Distribution of the scan statistic for a sequence of bistate trials, J. Appl. Prob. **38**, 908–916 (2001)
43.40 S. Kounias, M. Sfakianakis: The reliability of a linear system, its connection with the generalized birthday problem, Stat. Appl. **3**, 531–543 (1991)
43.41 R. A. Arratia, L. Gordon, M. S. Waterman: The Erdös Rényi Law in distribution for coin tossing, sequence matching, Ann. Stat. **18**, 539–570 (1990)
43.42 R. A. Arratia, M. S. Waterman: The Erdös Rényi strong law for pattern matching with a given proportion of mismatches, Ann. Prob. **17**, 1152–1169 (1989)
43.43 S. Karlin, F. Ost: Counts of long aligned word matches among random letter sequences, Adv. Appl. Prob. **19**, 293–251 (1987)
43.44 M. Mansson: On compound Poisson approximation for sequence matching, Comb. Prob. Comp. **9**, 529–548 (2000)
43.45 P. Deheuvels: On the Erdös Rényi theorem for random fields, sequences, its relationships with the theory of runs, spacings, Z. Wahrsch. **70**, 91–115 (1985)
43.46 R. A. Arratia, L. Goldstein, L. Gordon: Poisson approximation, the Chen–Stein method, Stat. Sci. **5**, 403–434 (1990)
43.47 S. Wallenstein, J. Naus, J. Glaz: Power of the scan statistic in detecting a changed segment in a Bernoulli sequence, Biometrika **81**, 595–601 (1995)
43.48 J. Glaz, J. Naus: Tight bounds, approximations for scan statistic probabilities for discrete data, Ann. Appl. Prob. **1**, 306–318 (1991)
43.49 M. V. Koutras: Consecutive k, r-out-of-n: DFM systems, Microelectronics Reliab. **37**, 597–603 (1997)
43.50 J. I. Naus: Probabilities for a generalized birthday problem, J. Am. Stat. Assoc. **69**, 810–815 (1974)

43.51 J. Krauth: Bounds for the upper tail probabilities of the circular ratchet scan statistic, Biometrics **48**, 1177–1185 (1992)

43.52 S. Wallenstein, C. R. Weinberg, M. Gould: Testing for a pulse in seasonal event data, Biometrics **45**, 817–830

43.53 J. Chen, J. Glaz: Approximations for discrete scan statistics on the circle, Stat. Prob. Lett. **44**, 167–176 (1999)

43.54 N. Nagarwalla: A scan statistic with a variable window, Stat. Med. **15**, 845–50 (1996)

43.55 J. Glaz, N. Balakrishnan: Spatial scan stat: Models, calculations and applications. In: *Scan Statistics and Applications*, ed. by M. Kulldorff (Birkhauser, Boston 1999) pp. 303–322

43.56 M. Kulldorff, G. Williams: *SatScan v. 1.0. Software for the Space and Space-Time Scan Statistics* (National Cancer Inst, Bethesda 1997)

43.57 J. I. Naus, S. Wallenstein: Multiple window, cluster size scan procedures, Method. Comp. Appl. Prob. **6**, 389–400 (2004)

43.58 N. H. Anderson, D. M. Titterington: Some methods for investigating spatial clustering with epidemiologic applications, J. R. Stat. Soc. **A160**, 87–105 (1997)

43.59 D. Q. Naiman, C. Priebe: Computing scan statistic p-values using importance sampling, with applications to genetics, medical image analysis, J. Comp. Graph. Stat. **10**, 296–328 (2001)

43.60 C. E. Priebe, D. Q. Naiman, L. M. Cope: Importance sampling for spatial scan analysis: computing scan statistic p-values for marked point processes, Comp. Stat. Data Anal. **35**, 475–485 (2001)

43.61 J. I. Naus: Clustering of random points in two dimensions, Biometrika **52**, 263–267 (1965)

43.62 C. Loader: Large deviation approximations to the distribution of scan statistics, Adv. Appl. Prob. **23**, 751–771 (1991)

43.63 I. P. Tu: Theory and Applications of Scan Statistics. Ph.D. Thesis (Stanford University, Palo-Alto 1997)

43.64 D. Aldous: *Probability Approximations via the Poisson Clumping Heuristic* (Springer, Berlin Heidelberg 1989)

43.65 S. E. Alm: On the distribution of scan statistics of a two-dimensional Poisson processes, Adv. Appl. Prob. **29**, 1–18 (1997)

43.66 S. E. Alm: Approximation, simulation of the distribution of scan statistics for Poisson processes in higher dimensions, Extremes **1**, 111–126 (1998)

43.67 G. Haiman, C. A. Preda: A new method for estimating the distribution of scan statistics for a two-dimensional Poisson Process, Method. Comp. Appl. Prob. **4**, 393–408 (2002)

43.68 M. D. Penrose: Focusing of the scan statistic and geometric clique number, Adv. Appl. Prob. **34**, 739–753 (2002)

43.69 D. Siegmund, B. Yakir: Tail probabilities for the null distribution of scanning stattistics, Bernoulli **6**, 191–213 (2000)

43.70 T. Akiba, T. Yamamoto, H. Yamamoto: Reliability of a 2-dimensional k-within-consecutive rxs out of mxn: F system.e=48, pages=625-637, year=2001,, Naval Res Log

43.71 R. W. R. Darling, M. S. Waterman: Extreme value distribution for the largest cube in a random lattice, SIAM J. Appl. Math. **46**, 118–132 (1986)

43.72 J. Chen, J. Glaz: Two dimensional discrete scan statistics, Stat. Prob. Lett. **31**, 59–68 (1996)

43.73 V. Karwe, J. Naus: New recursive methods for scan statistic probabilities, Comp. Stat. Data Anal. **23**, 389–402 (1997)

43.74 R. J. Huntington: Distribution of the minimum number of points in a scanning interval on the line, Stochastic Proc. Appl. **7**, 73–77 (1978)

43.75 J. Glaz, J. Naus: Multiple clusters on the line, Commun. Stat. Theory Meth. **12**, 1961–1986 (1983)

43.76 X. Su, S. Wallenstein: New approximations for the distribution of the *r*-scan statistic, Stat. Prob. Lett. **46**, 411–419 (2000)

43.77 X. Su, S. Wallenstein, D. Bishop: Non-overlapping clusters: Approximate distribution, application to molecular biology, Biometrics **57**, 420–426 (2001)

43.78 M. Roos: Compound Poisson approximations for the numbers of extreme spacings, Adv. Appl. Prob. **25**, 847–874 (1993)

43.79 J. Glaz, N. Balakrishnan: On Poisson Approximation for Continuous Multiple Scan Statistics in Two Dimensions. In: *Scan Statistics and Applications*, ed. by M. Mansson (Birkhauser, Boston 1999)

43.80 A. Godbole, S. G. Papastavrides: *Runs and Patterns in Probability* (Kluwer, Dordrecht 1994)

43.81 M. V. Boutsikas, M. V. Koutras: Modeling claim exceedances over thresholds, Insur. Math. Economics **30**, 67–83 (2002)

43.82 J. I. Naus, D. Wartenberg: A double scan statistic for clusters of two types of events, J. Am. Stat. Assoc. **92**, 1105–1113 (1997)

43.83 J. I. Naus, V. Stefanov: Double scan statistics, Method. Comp. Appl. Prob. **4**, 163–180 (2002)

43.84 G. P. Patil, C. Taille: Geographic, network surveillance via scan statistics for critical area detection, Stat. Sci. **18**, 457–465 (2003)

43.85 M. Kulldorff, Z. Fang, S. J. Walsh: A tree based scan statistic for database disease surveillance, Biometrics **59**, 323–331 (2003)

44. Condition-Based Failure Prediction

Machine reliability is improved if failures are prevented. Preventive maintenance (PM) can be performed in order to promote reliability, but only if failures can be predicted early enough. Although PM can be approached in different ways, according to time or condition, whichever one of these approaches is adopted, the key issue is whether a failure can be detected early enough or even predicted. This chapter discusses a way to predict failure, for use with PM, in which the state of a DC motor is estimated using the Kalman filter. The prediction consists of a simulation on a computer and an experiment performed on the DC motor. In the simulation, an exponential attenuator is placed at the output end of the motor model in order to simulate aging failure. Failure is ascertained by monitoring a state variables, the rotation speed of the motor. Failure times were generated by Monte Carlo simulation and predicted by the Kalman filter. One-step-ahead and two-steps-ahead predictions are performed. The resulting prediction errors are sufficiently small in both predictions. In the experiment, the rotating speed of the motor was measured every 5 min for 80 days. The measurements were used to perform Kalman prediction and to verify the prediction accuracy. The resulting prediction errors were acceptable. Decreasing the increment time between measurements was found to increase the accuracy of Kalman prediction uses. Consequently, it is shown that failure can be prevented (promoting reliability) by performing predictive maintenance depending on the results of state estimation using the Kalman filter.

Nomenclature

$(\cdot)_k$ = The value of $(\cdot)$ at time kT
$\widehat{(\cdot)}_{a/b}$ = The estimate of $(\cdot)$ at time aT based on all information known about the process up to time bT
$\boldsymbol{A}$ = A matrix
$\boldsymbol{A}_\mathrm{c}$ = Coefficient matrix of the state equation for a continuous system
$\boldsymbol{A}_\mathrm{d}$ = Coefficient matrix of the state equation for a discrete system
$\boldsymbol{A}^\mathrm{T}$ = Transpose matrix of $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$ = Inverse matrix of $\boldsymbol{A}$
B = Damping coefficient
$\boldsymbol{B}_\mathrm{c}$ = Coefficient matrix of the state equation for a continuous system
$\boldsymbol{B}_\mathrm{d}$ = Coefficient matrix of the state equation for a discrete system
$\boldsymbol{B}_\mathrm{k}$ = Coefficient matrix for the input term of a discrete state equation
$\boldsymbol{C}$ = A matrix
$\boldsymbol{C}_\mathrm{c}$ = Coefficient matrix of the state equation for a continuous system
$\boldsymbol{C}_\mathrm{d}$ = Coefficient matrix of the state equation for a discrete system

$\boldsymbol{D}_c$ = Coefficient matrix of the state equation for a continuous system
$\boldsymbol{D}_d$ = Coefficient matrix of the state equation for a discrete system
E = Applied voltage
E_r = Estimation error
$\boldsymbol{H}_k$ = Matrix giving the ideal (noiseless) connection between the measurement and the state vector
i_a = Armature winding current
J = Moment of inertia of rotor and load
k_b = Back emf constant
K_k = Kalman gain
k_T = Motor torque constant
L_a = Armature winding inductance
L^{-1} = The inverse Laplace transform
$\boldsymbol{P}_{k/k-1}$ = Estimation error covariance matrix
$\boldsymbol{Q}_k$ = Covariance matrices for disturbance
R = Armature winding resistance
$\boldsymbol{R}_k$ = Covariance matrices for noise
t = Time variable
T = Motor output torque
T = Increment time for every step in Kalman prediction
$\boldsymbol{U}_k$ = Control input of a discrete state equation at state k
V = Variation in the estimated rotating speed
$\boldsymbol{V}_k$ = Noise (measurement error vector), assumed to be a white sequence with known covariance
$\boldsymbol{W}_k$ = Disturbance (system stochastic input vector), assumed to be a white sequence with known covariance and zero cross-correlation with V_k sequence
x, X = Variable of a distribution function
$\boldsymbol{X}_{D0}$ = Initial states resulting from deterministic input
$\boldsymbol{X}_k$ = System state vector at state k
$\boldsymbol{X}_{S0}$ = Initial states resulting from stochastic input
$\boldsymbol{Y}_k$ = System output vector at state k
Z_k = Output measurement vector
θ = Motor angle displacement
$\dot{\theta}$ = Motor rotating speed
μ = Mean value of a distribution function
σ = Standard deviation of a distribution function
Φ_k = Matrix relating $\boldsymbol{X}_k$ to $\boldsymbol{X}_{k+1}$ in the absence of a forcing function, which is the state transition matrix if $\boldsymbol{X}_k$ is sampled from a continuous process

44.1 Overview

High quality and excellent performance of a system are always goals that engineers strive to achieve. Reliability engineering encourages system quality and performance from the beginning to the end of the system's lifecycle [44.1]. Therefore, reliability can be thought of as the time-dimensional quality of the system. Reliability is affected by every stage of the system's lifecycle, including its development, design, production, quality control, shipping, installation, operation, and maintenance. Consequently, paying attention to each of the stages promotes reliability. Specifically, in the onsite operation phase, failures are the main causes of worsened performance and degraded reliability. This is a very important consideration for any equipment that may cause severe damage to public safety or financial benefit upon failure, such as nuclear power plants, passenger vehicles, or semiconductor production lines. Accordingly, failure avoidance is the main approach to ensuring reliability. Effective maintenance is the best way to reduce failure [44.1]. There are three main types of maintenance: improvement maintenance (IM), corrective maintenance (CM), and preventive maintenance (PM) [44.2]. The purpose of IM is to reduce the need for maintenance or eliminate the need for it entirely . Therefore IM should be performed at the design phase of a system in order to emphasize the elimination of failure. There are many restrictions on a designer, however, such as space, budget, and market requirements. Usually the reliability of a product is related to its price.

On the other hand, CM is the repair performed after failure occurs, while PM refers to all of the actions intended to maintain equipment in good operating condition and to avoid failure [44.2]. The most common strategy for maintenance is scheduled maintenance, where maintenance occurs at set times, at set operational times, after set amounts of material have been processed, or by some other prescribed criteria. Nevertheless, there are at least two drawbacks to this type of maintenance:

1. The criteria on which the scheduled maintenance is based are statistical averages, such as mean time to failure (MTTF). This leads to an unavoidable risk that a system can fail before the criteria are exceeded; in other words a failure may occur unexpectedly.
2. The real duty cycles for certain parts or modules may be longer than those averages, but they are still replaced during a scheduled maintenance, which is wasteful.

In contrast, condition monitoring is a more reasonable type of maintenance than scheduled maintenance. However, a failure really needs to be detected prior to its occurrence.

The relationship between error, failure, and fault is illustrated in Fig. 44.1, and the three terms are defined as follows [44.3]:

1. An error is a discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition.
2. Failure is an event where a required function is terminated (exceeding the acceptable limits).
3. Fault is the state characterized by an inability to perform a required function, excluding the inability encountered during preventive maintenance or other planned actions, or due to lack of external resources.

Based on the above statements, an error is not a failure and a fault is hence a state resulting from a failure. An error is sometimes referred to as an incipient failure [44.4]. Therefore, PM action is taken when the system is still at an error condition – within acceptable deviation and before failure occurs. Hence, PM is an effective approach to promoting reliability [44.5]. As mentioned before, time-based and condition-based maintenance are two major approaches to PM. Irrespective of the approach adopted for PM, the key issue is whether a failure can be detected early or even predicted.

Many methods have been proposed for failure prediction, such as statistical knowledge of the reliability parameters [44.6, 7], neural network studies [44.8], and understanding the failure mechanism of damaged products [44.9]. Fault detection based on modeling and estimation is one of these methods [44.10]. The Kalman filter is useful not only for state estimation but also for state prediction. It has been widely used in different fields over the past few decades, such as in on-line failure detection [44.11], real-time prediction of vehicle motion [44.12], and prediction of maneuvering target trajectories [44.13]. The Kalman filter is a linear, discrete-time, and finite-dimensional system [44.14]. Its appearance is a copy of the system that is estimated. Inputs to the filter include the control signal and the difference value between measured and estimated state variables. Actual values of the event acquired by the monitor sensor are fed into the corresponding Kalman filter to execute state estimation. By minimizing mean-square estimation errors, an optimal estimate can be derived. Using the current state, the Kalman filter provides a predicted value of the next state for the corresponding event at every time interval T. As a result, the output of the filter becomes optimal estimates of the state variables for the next time step. Each event has a prescribed failure threshold, and the predicted value is compared with the prescribed failure threshold in order to judge whether the monitored event has failed after T or is still within the established threshold. Once the estimated value reaches the threshold, failure is predicted. Therefore, the current state is a warning state and PM needs to be performed. If predicted state variables indicate that a device is going to fail, then the failure can be prevented in time using PM. However, future state variables need to be accurately predicted a reasonable time before failure occurrence [44.10, 15].

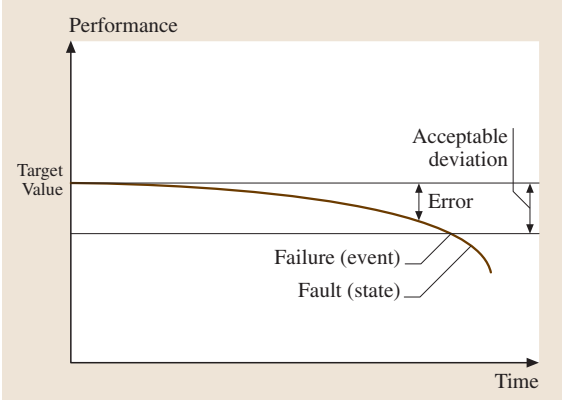


Fig. 44.1 Error, failure and fault

This chapter is about state estimation and how to predict the need for PM using a Kalman filter. A DC motor is the object on which we perform condition-based failure predictions. The prediction consists of two parts. The first part is a simulation on a computer, and the second part is an experiment on the DC motor. In the simulation, failure times were generated by Monte Carlo simulation (MCS) and predicted by the Kalman filter. One-step-ahead and two-steps-ahead predictions were conducted. The resulting prediction errors are sufficiently small in both predictions. Even so, the failure prediction was still simulated on a computer. In the second part of the prediction, a DC motor and a data acquisition system are used to implement the simulation. The rotation speed of the motor is chosen as the major state variable to judge whether the motor is going to fail, by estimating state using the Kalman filter. The rotation speed of the motor was measured and recorded every 5 min for 80 d. Instead of simulated data, the measured data are used to perform Kalman prediction and to verify the prediction accuracy.

In Sect. 44.2, we study a discrete system model with deterministic control input, white noise disturbance

and noisy output measurement. We then formulate an equation for state estimation of the Kalman filter. Deterministic inputs are considered in this formulation. Moreover, equations for N-step-ahead prediction are derived. Section 44.3 presents the transfer function, the continuous state model, and the discrete state model of the DC motor that is employed as an example in this chapter. Section 44.4 presents a simulated system with prescribed parameters, a Monte Carlo simulation and an ARMA model used to generate necessary data for a failure prediction simulation, and the exponential attenuator used to simulate the aging failure mode. Simulation results are also given and discussed. Section 44.5 presents the experimental set-up with related parameters, experimental results and discussion. Section 44.6 concludes this work.

44.2 Kalman Filtering

This section introduces the Kalman filtering that is used in this chapter to perform the failure prediction.

44.2.1 System Model

The block diagram of a discrete system is shown in Fig. 44.2. The state equations [44.14] are:

$$\boldsymbol{X}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{X}_k + \mathbf{B}_k \boldsymbol{U}_k + \boldsymbol{W}_k \,, \tag{44.1}$$

$$\boldsymbol{Y}_k = \mathbf{H}_k \boldsymbol{X}_k \,, \tag{44.2}$$

$$\boldsymbol{Z}_k = \boldsymbol{Y}_k + \boldsymbol{V}_k \,. \tag{44.3}$$

Substituting (44.2) into (44.3) yields

$$\boldsymbol{Z}_k = \mathbf{H}_k \boldsymbol{X}_k + \boldsymbol{V}_k \,. \tag{44.4}$$

Let $E[\boldsymbol{X}]$ be the expected value of $\boldsymbol{X}$; therefore, the covariance matrices for $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ are given by:

$$E\left[\boldsymbol{W}_k \boldsymbol{W}_i^{\mathrm{T}}\right] = \begin{cases} \mathbf{Q}_k \,, & i = k \\ 0 \,, & i \neq k \end{cases} \tag{44.5}$$

$$E\left[\boldsymbol{V}_k \boldsymbol{V}_i^{\mathrm{T}}\right] = \begin{cases} \mathbf{R}_k \,, & i = k \\ 0, & i \neq k \end{cases} \tag{44.6}$$

$$E\left[\boldsymbol{W}_k \boldsymbol{V}_i^{\mathrm{T}}\right] = 0 \,, \quad \text{for all } k \text{ and } i \,. \tag{44.7}$$

It follows that both $\mathbf{Q}_k$ and $\mathbf{R}_k$ are symmetric and positive definite [44.16].

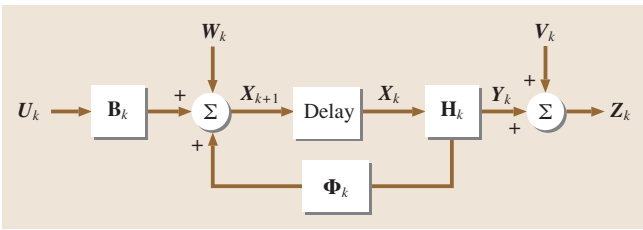


Fig. 44.2 Block diagram of a discrete system

44.2.2 State Estimation

State estimation involves guessing the value of $\boldsymbol{X}_k$ by using measured data, i. e. Z_0, Z_1, ... Z_{k-1}. Accordingly, $\hat{\boldsymbol{X}}_{k/k-1}$ is called the prior estimate of $\boldsymbol{X}$, and $\hat{\boldsymbol{X}}_{k/k}$ is called the posterior estimate of $\boldsymbol{X}$ [44.16]. The prior estimation error is defined as

$$\mathbf{e}_{k/k-1} = \boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k-1} \,. \tag{44.8}$$

Since $\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ are assumed to be white sequences, the prior estimation error has a mean of zero. Consequently, the associated error covariance matrix is written as

$$\begin{aligned} \boldsymbol{P}_{k/k-1} &= E\left[(\mathbf{e}_{k/k-1})(\mathbf{e}_{k/k-1})^{\mathrm{T}}\right] \\ &= E\left[(\boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k-1})(\boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k-1})^{\mathrm{T}}\right] . \end{aligned} \tag{44.9}$$

The estimation problem begins with no prior measurements. Thus, the stochastic portion of the initial estimate is zero if the stochastic process mean is zero; i. e. $\hat{\boldsymbol{X}}_{0/-1}$ is only driven by deterministic input X_{D0}. It follows from (44.8) that

$$e_{0/-1} = X_0 - \hat{\boldsymbol{X}}_{0/-1} = X_0 - X_{D0} = X_{S0} \,. \tag{44.10}$$

Employing (44.9) and (44.10) yields

$$P_{0/-1} = E\left[\boldsymbol{X}_{S0} \boldsymbol{X}_{S0}^{\mathrm{T}}\right] \,. \tag{44.11}$$

The Kalman filter is a copy of the original system and is driven by the estimation error and the deterministic input. The block diagram of the filter structure is shown in Fig. 44.3. The filter is used to improve the prior estimate to make it the posterior estimate via the measurement $\boldsymbol{Z}_k$. A linear blending of the noisy measurement and the prior estimate is written as [44.16]

$$\hat{\boldsymbol{X}}_{k/k} = \hat{\boldsymbol{X}}_{k/k-1} + \boldsymbol{K}_{\mathrm{k}}(\boldsymbol{Z}_k - \boldsymbol{H}_k \hat{\boldsymbol{X}}_{k/k-1}) \,, \tag{44.12}$$

where K_{k} is the blending factor for this structure. Once the posterior estimate is determined, the posterior estimation error and associated error covariance matrix can

be derived as

$$\mathbf{e}_{k/k} = \boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k} \,, \tag{44.13}$$

$$\begin{aligned}\boldsymbol{P}_{k/k} &= E[(\mathbf{e}_{k/k})(\mathbf{e}_{k/k})^T] \,, \\ &= E[(\boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k})(\boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k})^T] \,. \end{aligned} \tag{44.14}$$

The optimal blending factor is written as [44.16]

$$K_{\mathrm{k}} = \boldsymbol{P}_{k/k-1}\mathbf{H}_{\mathrm{k}}^{\mathrm{T}}(\mathbf{H}_k\boldsymbol{P}_{k/k-1}\mathbf{H}_{\mathrm{k}}^{\mathrm{T}} + \mathbf{R}_k)^{-1} \,. \tag{44.15}$$

This specific K_{k}, namely the one that minimizes the mean-square estimation error, is called the Kalman gain.

Substituting (44.15) into (44.12), the posterior error covariance matrix can be derived as follows:

$$\begin{aligned}\boldsymbol{P}_{k/k} &= \boldsymbol{P}_{k/k-1} - \boldsymbol{P}_{k/k-1}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{H}_k\boldsymbol{P}_{k/k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right)^{-1} \\ &\quad \times \mathbf{H}_k\boldsymbol{P}_{k/k-1} \\ &= \boldsymbol{P}_{k/k-1} - \boldsymbol{K}_{\mathrm{k}}\left(\mathbf{H}_k\boldsymbol{P}_{k/k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right)\boldsymbol{K}_{\mathrm{k}}^{\mathrm{T}} \\ &= (I - \boldsymbol{K}_{\mathrm{k}}\mathbf{H}_k)\boldsymbol{P}_{k/k-1} \,. \end{aligned} \tag{44.16}$$

As depicted in Fig. 44.3, the one-step-ahead estimate is formulated as

$$\begin{aligned}\hat{\boldsymbol{X}}_{k+1/k} &= \boldsymbol{\Phi}_k\hat{\boldsymbol{X}}_{k/k-1} + \boldsymbol{\Phi}_k\boldsymbol{K}_{\mathrm{k}}(Z_k - \boldsymbol{H}_k\hat{\boldsymbol{X}}_{k/k-1}) \\ &\quad + \boldsymbol{B}_k\boldsymbol{U}_k \\ &= \boldsymbol{\Phi}_k[\hat{\boldsymbol{X}}_{k/k-1} + \boldsymbol{K}_{\mathrm{k}}(\boldsymbol{Z}_k - \boldsymbol{H}_k\hat{\boldsymbol{X}}_{k/k-1})] \\ &\quad + \boldsymbol{B}_k\boldsymbol{U}_k \\ &= \boldsymbol{\Phi}_k\hat{\boldsymbol{X}}_{k/k} + \boldsymbol{B}_k\boldsymbol{U}_k \,. \end{aligned} \tag{44.17}$$

Consequently, the one-step-ahead estimation error is derived as

$$\begin{aligned}\mathbf{e}_{k+1/k} &= (\boldsymbol{\Phi}_k\boldsymbol{X}_k + \mathbf{B}_k\boldsymbol{U}_k + \mathbf{W}_k) - (\boldsymbol{\Phi}_k\hat{\boldsymbol{X}}_{k/k} + \mathbf{B}_k\boldsymbol{U}_k) \\ &= \boldsymbol{\Phi}_k(\boldsymbol{X}_k - \hat{\boldsymbol{X}}_{k/k}) + \mathbf{W}_k \\ &= \boldsymbol{\Phi}_k\mathbf{e}_{k/k} + \mathbf{W}_k \,. \end{aligned} \tag{44.18}$$

In a manner similar to (44.14), the one-step-ahead error covariance matrix is derived as

$$\begin{aligned}\boldsymbol{P}_{k+1/k} &= E\left[(\boldsymbol{\Phi}_k\mathbf{e}_{k/k} + \mathbf{W}_k)(\boldsymbol{\Phi}_k\mathbf{e}_{k/k} + \mathbf{W}_k)^{\mathrm{T}}\right] \\ &= \Phi_k\boldsymbol{P}_{k/k}\Phi_k^{\mathrm{T}} + \boldsymbol{Q}_k \,. \end{aligned} \tag{44.19}$$

According to the above statements, we can make several remarks about Kalman estimation:

1. Since $\boldsymbol{K}_{\mathrm{k}}$ is optimal, the posterior estimate $\hat{\boldsymbol{X}}_{k/k}$ is an optimal estimate.
2. Based on (44.12), (44.15), (44.16), (44.17) and (44.19), recursive steps for constructing a one-step estimator are summarized in Fig. 44.4.
3. The recursive loop involves two different kinds of updating. Equations (44.12) and (44.16) yielding $\hat{\boldsymbol{X}}_{k/k}$ and $\boldsymbol{P}_{k/k}$ from $\hat{\boldsymbol{X}}_{k/k-1}$ and $\boldsymbol{P}_{k/k-1}$ are measurement updates; Equations (44.17) and (44.19) projecting $\hat{\boldsymbol{X}}_{k/k}$ and $\boldsymbol{P}_{k/k}$ to $\hat{\boldsymbol{X}}_{k+1/k}$ and $P_{k+1/k}$ are time updates.
4. The initial conditions, i. e. $\hat{\boldsymbol{X}}_{0/-1}$, $P_{0/-1}$, Φ_0, H_0, Q_0, and R_0 have to be known before we can start the recursive steps.

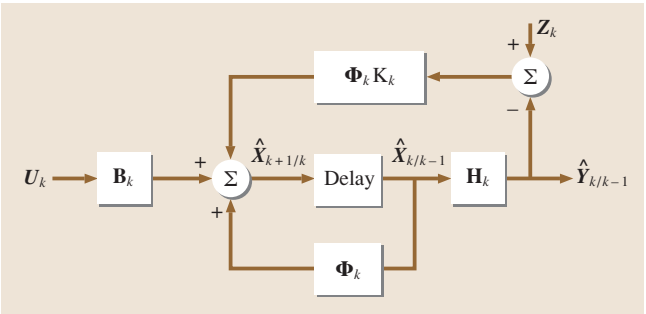


Fig. 44.3 Block diagram of a Kalman filter

44.2.3 Prediction

The estimate resulting from the recursive steps in Fig. 44.4 is a one-step-ahead prediction. Based on the posterior estimate (44.12), the state that is N steps ahead of the measurement $\boldsymbol{Z}_k$ can be predicted by using the ARMA (autoregressive and moving average) model [44.16]. From (44.17) and (44.19), the equations for N-step-ahead prediction are derived as

$$\begin{aligned}\hat{\boldsymbol{X}}_{k+N/k} &= \left(\prod_{i=k+N-1}^{k} \Phi_i\right)\hat{\boldsymbol{X}}_{k/k} \\ &\quad + \sum_{m=k}^{k+N-2}\left[\left(\prod_{i=k+N-1}^{m+1}\Phi_i\right)\boldsymbol{B}_m\boldsymbol{U}_m\right] \\ &\quad + \boldsymbol{B}_{k+N-1}\boldsymbol{U}_{k+N-1} \,, \end{aligned} \tag{44.20}$$

$$\begin{aligned}P_{k+N/k} &= \left(\prod_{i=k+N-1}^{k}\Phi_i\right)\boldsymbol{P}_{k/k}\left(\prod_{j=k}^{k+N-1}\Phi_j^{\mathrm{T}}\right) \\ &\quad + \sum_{m=k}^{k+N-2}\left[\left(\prod_{i=k+N-1}^{m+1}\Phi_i\right)Q_m \right. \\ &\quad \left. \times \left(\prod_{j=m+1}^{k+N-1}\Phi_j^{T}\right)\right] + Q_{k+N-1} \,. \end{aligned} \tag{44.21}$$

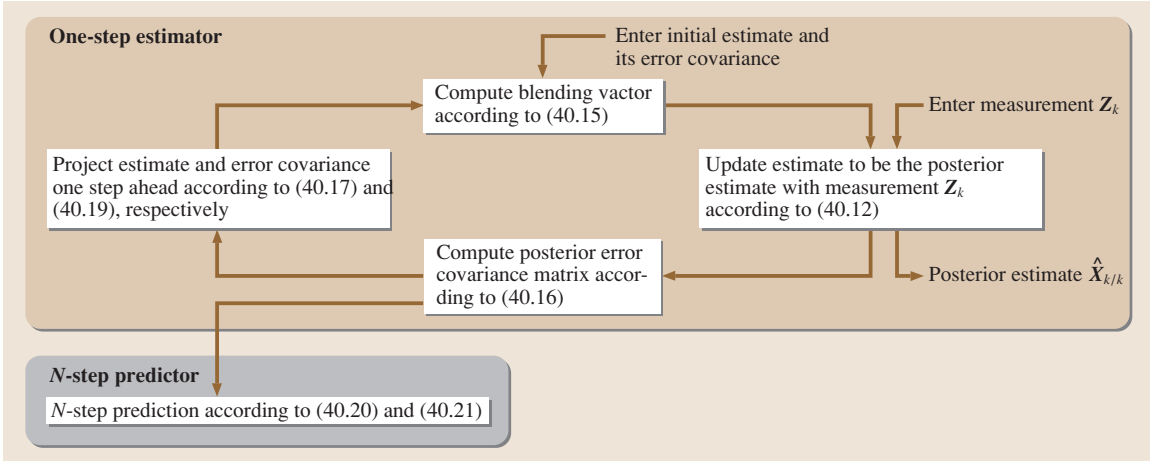


Fig. 44.4 One-step estimator and N-step predictor

The N-step predictor is an appendage of the one-step estimation loop [44.16]. It is also shown in Fig. 44.4. Since the current predicted value is assumed to be the initial value for the next prediction, the more steps the predictor predicts, the larger the error.

44.3 Armature-Controlled DC Motor

An armature-controlled DC motor is employed in this section as the physical model on which to perform failure prediction. The motor circuit representation is shown in Fig. 44.5.

44.3.1 Transfer Function

Using the properties of a DC motor, the following equations can be formulated [44.17]:

$$\phi = k_f i_f \,, \tag{44.22}$$

$$\begin{aligned} T &= \frac{ZP}{2\pi a}\phi i_a \\ &= k_1 \left(k_f i_f\right) i_a \\ &= k_T i_a \end{aligned} \tag{44.23}$$

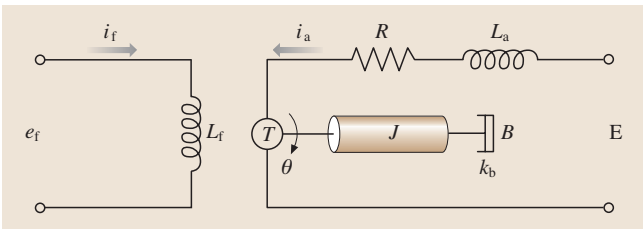


Fig. 44.5 Circuit representation of DC motor

$$e_b = k_b \frac{d\theta}{dt} \,, \tag{44.24}$$

$$L_a \frac{d}{dt} i_a + R i_a + e_b = E \,, \tag{44.25}$$

$$J\ddot{\theta} + B\dot{\theta} = T \,, \tag{44.26}$$

where $k_1 = \frac{ZP}{2\pi a}$ is called the motor constant, and $k_T = k_1 \left(k_f i_f\right)$ is the motor torque constant.

Taking the Laplace transform for (44.24), (44.25) and (44.26) results in

$$E_b(s) = k_b s\theta(s) \,, \tag{44.27}$$

$$\left(L_a s + R\right) I_a(s) = E(s) - E_b(s) \,, \tag{44.28}$$

$$\left(Js^2 + Bs\right)\theta(s) = T(s) = k_T I_a(s) \,. \tag{44.29}$$

Combining (44.27), (44.28) and (44.29), the transfer function of a DC motor is derived as

$$\frac{\theta(s)}{E(s)} = \frac{k_T}{s\left[\left(sL_a + R\right)\left(sJ + B\right) + k_T k_b\right]} \,. \tag{44.30}$$

Accordingly, the block diagram of a DC motor is as shown in Fig. 44.6. If $L_a \approx 0$, (44.30) can be rewritten as

$$\frac{\theta(s)}{E(s)} = \frac{k_m}{s\left(s\tau_m + 1\right)} \,, \tag{44.31}$$

where $k_m = \frac{k_T}{RB+k_T k_b}$ and $\tau_m = \frac{RJ}{RB+k_T k_b}$ are called the motor gain constant and the motor time constant, respectively.

44.3.2 Continuous State Space Model

Define θ, $\dot{\theta}$, and i_a as state variables, so that the state vector is $\boldsymbol{X} = [\theta\, \dot{\theta}\, i_a]^T$.

Since

$$\frac{d}{dt}\theta = \dot{\theta}\,, \tag{44.32}$$

substituting (44.23) and (44.32) into (44.26) yields

$$\frac{d}{dt}\dot{\theta} = \frac{1}{J}\left(k_T i_a - B\dot{\theta}\right) = \frac{k_T}{J} i_a - \frac{B}{J}\dot{\theta}\,. \tag{44.33}$$

Moreover, substituting (44.24) into (44.25) yields

$$\frac{d}{dt} i_a = \frac{1}{L_a}\left(E - R i_a - e_b\right) = \frac{E}{L_a} - \frac{k_b}{L_a}\dot{\theta} - \frac{R}{L_a} i_a\,. \tag{44.34}$$

During measurement, the rotation speed $\dot{\theta}$ is the motor output. According to (44.32), (44.33) and (44.34), the continuous state equations of the DC motor are

$$\frac{d}{dt}\begin{pmatrix}\theta\\ \dot{\theta}\\ i_a\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\ 0 & -\frac{B}{J} & \frac{k_T}{J}\\ 0 & -\frac{k_b}{L_a} & -\frac{R}{L_a}\end{pmatrix}\begin{pmatrix}\theta\\ \dot{\theta}\\ i_a\end{pmatrix} + \begin{pmatrix}0\\ 0\\ \frac{1}{L_a}\end{pmatrix} E, \tag{44.35}$$

$$Y = \begin{pmatrix}0 & 1 & 0\end{pmatrix}\begin{pmatrix}\theta\\ \dot{\theta}\\ i_a\end{pmatrix}\,. \tag{44.36}$$

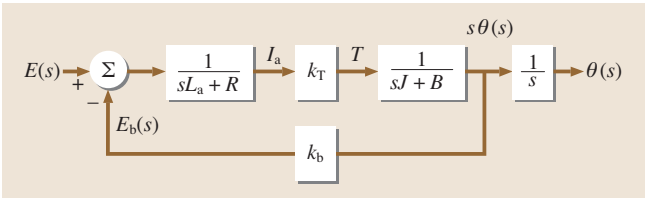


Fig. 44.6 Block diagram of DC motor

44.3.3 Discrete State Space Model

The state equations for a continuous system take the form of [44.18]:

$$\begin{aligned}\dot{X}(t) &= A_c X(t) + B_c U(t)\\ Y(t) &= C_c X(t) + D_c U(t)\end{aligned} \tag{44.37}$$

Let $\Phi_c(t) = L^{-1}[(s\mathbf{I} - \mathbf{A}_c)^{-1}]$ be the state transition matrix for (44.37). The discrete state equations sampled from (44.37) by a sample-and-hold with a time interval of T seconds are as follows [44.19]:

$$\begin{aligned}\boldsymbol{X}_{k+1} &= \mathbf{A}\boldsymbol{X}_k + \mathbf{B}\boldsymbol{U}_k\,,\\ \boldsymbol{Y}_k &= \mathbf{C}\boldsymbol{X}_k + \mathbf{D}\boldsymbol{U}_k\,,\end{aligned}$$

where

$$\mathbf{A} = \Phi_c(T)\,, \tag{44.38}$$

$$\mathbf{B} = \left[\int_0^T \Phi_c(\tau)\,d\tau\right] B_c\,, \tag{44.39}$$

$$\boldsymbol{C} = \boldsymbol{C}_c\,, \tag{44.40}$$

$$\boldsymbol{D} = \boldsymbol{D}_c\,. \tag{44.41}$$

44.4 Simulation System

This section describes a computer simulation of failure prediction for PM on a DC motor.

44.4.1 Parameters

The parameters used for the DC motor in this simulation are: follows [44.20]:

$E = 10\,\mathrm{V}$,
$B = 0.001\,\mathrm{N\,m\,s}$,
$J = 0.01\,\mathrm{kg\,m^2}$,
$K_T = 1\,\mathrm{N\,m\,/\,A}$,
$K_b = 0.02\,\mathrm{V\,s}$,
$R = 10\,\Omega$,
$L_a = 0.01\,\mathrm{H}$.

Substituting them into (44.35) and (44.36), the continuous state equations of the motor become

$$\frac{d}{dt}\begin{pmatrix}\theta\\ \dot{\theta}\\ i_a\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\ 0 & -0.1 & 100\\ 0 & -2 & -1000\end{pmatrix}\begin{pmatrix}\theta\\ \dot{\theta}\\ i_a\end{pmatrix} + \begin{pmatrix}0\\ 0\\ 100\end{pmatrix} 10\,, \tag{44.42}$$

$$Y = \begin{pmatrix}0 & 1 & 0\end{pmatrix}\begin{pmatrix}\theta\\ \dot{\theta}\\ i_a\end{pmatrix}\,. \tag{44.43}$$

The following parameters are also used to predict failure:

1. The failure threshold of the motor is defined as 5% less than the normal value, which is set to be the

initial estimate in the Kalman prediction procedure. That is, the motor is judged to fail if the rotation speed drops to 95% of the normal value.
2. Mean time between failure (MTBF) for the motor is 100 000 h [44.21].
3. Sampling interval T is 1 h (this is the increment time for every step in Kalman prediction).
4. Disturbance $\boldsymbol{W}_k$ has mean 0 and variance 0.01 V [44.22].
5. Measurement error $\boldsymbol{V}_k$ for $\dot{\theta}$ has a zero mean and a standard deviation of 3.333 rad/s, which is 1% of the full-scale accuracy [44.23] of the measurement.
6. PM lead-time is set to $n \times 60$ min, where n is the number of steps ahead in the prediction. Accordingly, the alarm signal is activated (to indicate that PM should be executed) whenever the Kalman filter predicts that the motor speed will be lower than the prescribed threshold $n \times 60$ min later.

44.4.2 Monte Carlo Simulation and ARMA Model

Assuming that motor failures occur randomly, a Monte Carlo simulation (MCS) can be used to generate the failure times of the motor. The relationship between the failure rate $h(t)$ and the distribution function of the lifetime $f(t)$ is [44.5]

$$f(t) = h(t) \exp\left[-\int_0^t h(\tau)\,\mathrm{d}\tau\right] . \qquad (44.44)$$

Failures occur randomly during the useful lifetime, statistically conforming to a bathtub curve [44.5]. The failure rate is constant during this period. Let the failure rate in (44.44) be a constant λ, and so (44.44) becomes

$$f(t) = \lambda \exp\left(-\int_0^t \lambda\,\mathrm{d}\tau\right) = \lambda\,\mathrm{e}^{-\lambda t} , \qquad (44.45)$$

which is an exponential distribution function. Let u_i, $i = 1, 2, 3, \ldots, m$, represent a set of standard uniformly distributed random numbers. The corresponding numbers t_i of the random variable t in (44.45) (in other words the simulated failure times), are written as [44.5]

$$t_i = -\frac{1}{\lambda} \ln u_i , \qquad (44.46)$$

with exponential distributions.

The measurements needed for the recursive estimation loop of the Kalman filter, as depicted in Fig. 44.4, are generated by the ARMA model ((44.1) to (44.3)). Simulations in this section are performed using MATLAB [44.24]. All random numbers and white sequences with prescribed variances needed are obtained using the random number generator in MATLAB.

44.4.3 Exponential Attenuator

To account for the aging failure modes and the exponentially distributed failure times t_i, an exponential attenuator, represented by $\mathrm{e}^{-t/\tau}$, is placed at the output ends of both the motor system and the Kalman filter. The block diagram of the simulation system is shown in Fig. 44.7. The symbol τ of the attenuator in Fig. 44.7 denotes the failure time constant of the motor, which varies with the failure times that are generated by the MCS.

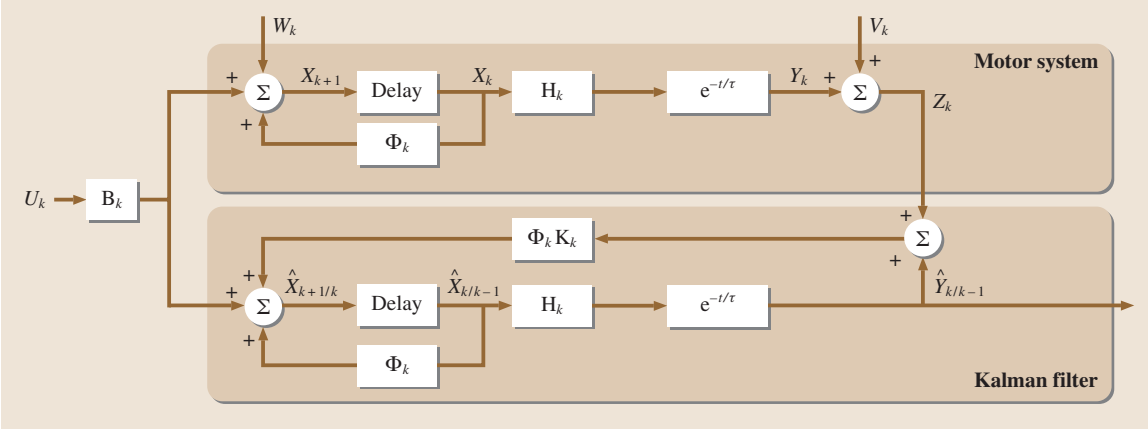


Fig. 44.7 Block diagram of the simulation system

44.4.4 Simulation Results

Two categories of simulation are conducted in this section, namely one-step-ahead prediction and two-steps-ahead prediction. According to the central limit theorem (CLT), estimators follow the normal distribution if the sample size is sufficiently large. A sample size of 30 is a reasonable number to use [44.25]. The larger the sample size, the smaller the estimated error, which tends to zero when the sample size approaches infinity. Hence, each simulation is executed 100 times. Simulation results for a lead-time of 60 min – one-step-ahead prediction – is shown in Fig. 44.8. Figure 44.8a shows the results for the 100 simulations of failure times generated by the MCS, the failure times predicted by the Kalman filter, and the associated alarm times. Figure 44.8b shows the results from one of the 100 simulations with properly scaled coordinates. The failure time differences between MCS and Kalman prediction are shown in Fig. 44.9. The mean value and the standard deviation of the differences in the 100 simulations are −34.71 min and 65.90 min, respectively. The negative sign of the mean value indicates that the failure time predicted by the Kalman filter occurs earlier than the time generated by MCS. According to the Z formula [44.25], the error in estimating the mean value of the sample population can be calculated by

$$E_r^2 = \frac{Z_{\alpha/2}^2 \sigma^2}{n} .$$

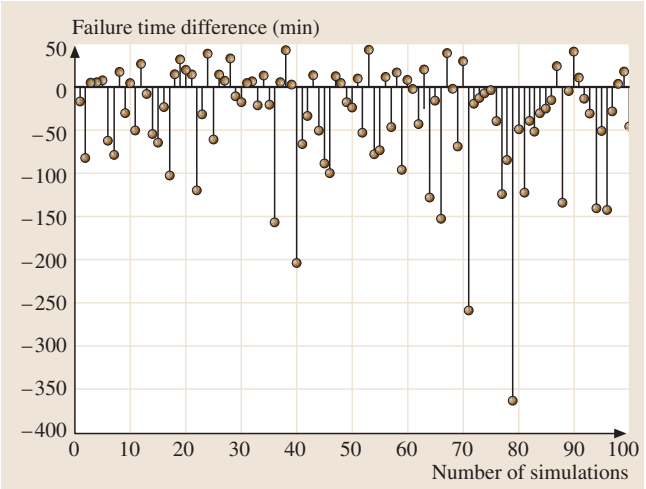


Fig. 44.9 Differences between the failure times given by Monte Carlo simulation results and Kalman filter predictions when the lead-time = 60 min

The Z value for a 99% confidence level is 2.575 [44.25]. Solving for E_r gives

$$E_r = \frac{(2.575)(65.8954)}{\sqrt{100}} = 16.97(\text{min}) .$$

According to the above data, we can say with 99% confidence that the mean value of the time difference between the MCS and the Kalman prediction is -34.71 ± 16.97 min, in other words from −17.74 min to −51.68 min. Taking the time difference into account,

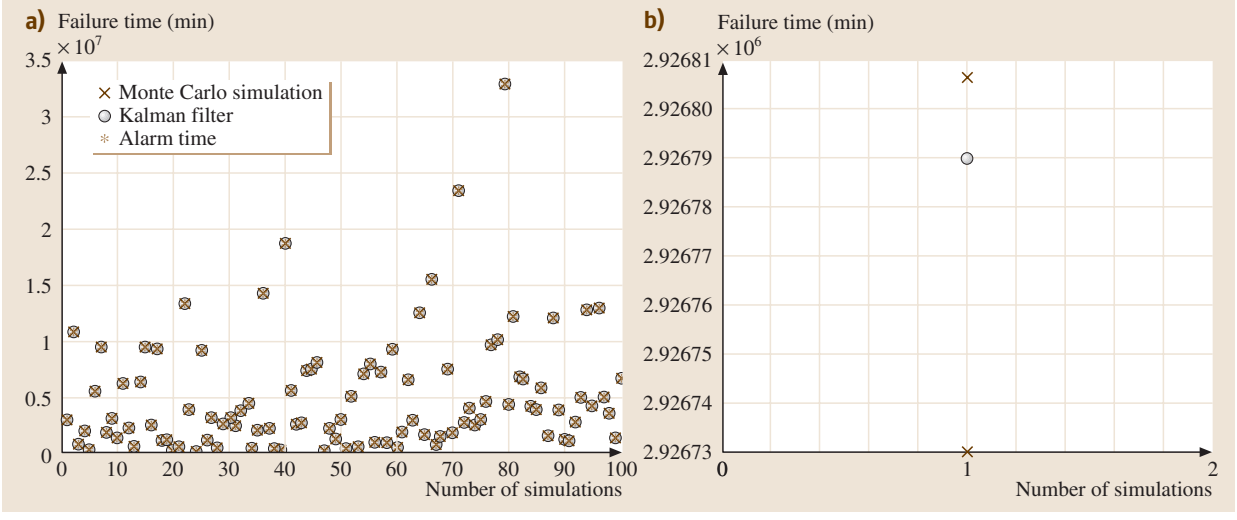


Fig. 44.8 Failure times generated by Monte Carlo simulation and predicted by Kalman filter when lead-time = 60 min

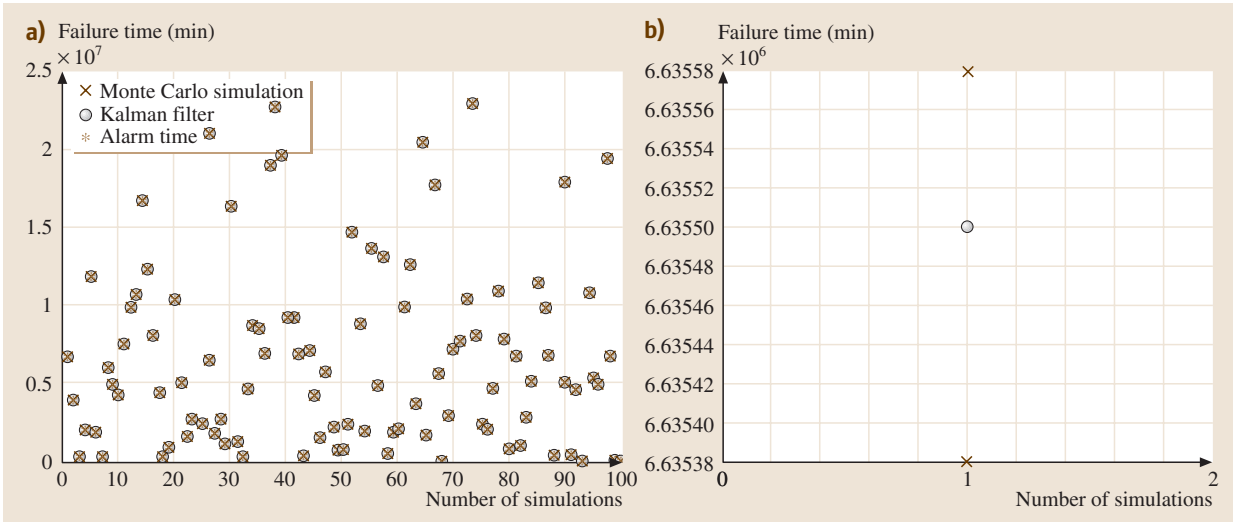


Fig. 44.10 Failure times generated by Monte Carlo simulation and predicted by Kalman filter when the lead-time = 120 min

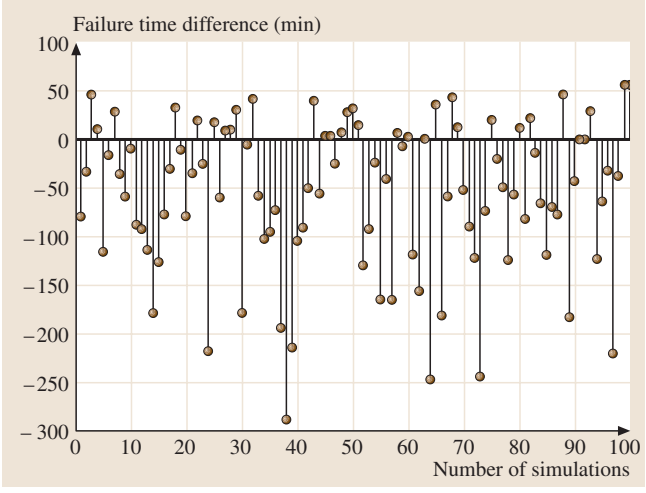


Fig. 44.11 Differences between the failure times given by Monte Carlo simulation results and Kalman filter predictions when the lead-time = 120 min

the alarm signal will appear at least 77.74 min prior to failure occurrence.

The results for the second category of simulation, – two-steps-ahead prediction, with a PM lead-time of 120 min – are shown in Figs. 44.10 and 44.11. The mean value of the difference in failure times from the MCS and the Kalman prediction is −56.34 min, and the 99% confidence interval for this mean is 20.06 min. The maximum prediction error for this case is 76.40 min, which is 1.48 times greater than the error from the one-step-ahead prediction.

44.4.5 Notes About the Simulation

1. In order to avoid false alarms, the failure threshold should not be set too close to the normal value. Otherwise, a decision-making algorithm is needed to identify that a failure has actually been predicted.
2. The disturbance amplitude should comprise all possible uncertainties about the motor and the environment.
3. The proposed method cannot deal with abrupt changes during a sampling interval. Thus, the sampling interval should not be too long.
4. Since the prediction is made for purposes of PM, the prediction time should be long enough that PM can be performed before the failure occurs.
5. In contrast to the deterministic portion, the variance that is driven by the disturbance of the system is small. The difference in state variables between prediction steps fades very rapidly. Thus, using the N-step predictor ((44.20)), only the prediction for the first few steps is of significance.
6. The proposed method in this section is exemplified by a motor system, which is treated as a component. The procedure can be executed on a multicomponent system if state equations can be constructed for the components as a whole. The procedure can be performed on either the multicomponent system or each of the components.

For a complicated or large system, the proposed method can be only performed on those elements that are present in minimum cut sets as constructed by fault tree analysis or a Petri net model for failure [44.26].

7. Multiple failure modes can be modeled as modules, such as an attenuator for simulating an aging failure mode for the electrical motor exemplified in this paper, and placed at the system model output end to extend the proposed method. As described previously, the system model may be single-component or multicomponent. System failure analysis can determine whether the failure modules occur in series, parallel or in other arrangements [44.26]. As for a multicomponent system with multiple failure modes, the system can be split into several components and the related failure module(s) placed at the output end of each component in order to perform state estimation using the Kalman filter for each component.

44.5 Armature-Controlled DC Motor Experiment

This section presents a PM failure-prediction experiment for a DC motor.

44.5.1 Experiment Design

The experiment design, shown in Fig. 44.12, comprises a DC motor with a driver unit and a data acquisition system.

DC Motor

The DC motor used in this experiment is made by TECO, Taiwan. The model number of the motor is GSDT-1/2 hp. Parameters for the DC motor used in this study are as follows [44.27]:

$E = 150\,\mathrm{V}$, $B = 0.001\,135\,\mathrm{N\,m\,s}$, $J = 0.0102\,\mathrm{kg\,m^2}$, $K_\mathrm{T} = 0.153\,\mathrm{N\,m/A}$, $K_\mathrm{b} = 1.926\,\mathrm{V\,s}$, $R = 3.84\,\Omega$, $L_\mathrm{a} = 0.01\,\mathrm{H}$.

Substituting these into (44.35) and (44.36), the continuous state equations of the motor become

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\theta\\ \dot\theta\\ i_\mathrm{a}\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\ 0 & -0.111 & 15\\ 0 & -192.6 & -384\end{pmatrix}\begin{pmatrix}\theta\\ \dot\theta\\ i_\mathrm{a}\end{pmatrix} + \begin{pmatrix}0\\ 0\\ 100\end{pmatrix}150\,, \tag{44.47}$$

$$Y = \begin{pmatrix}0 & 1 & 0\end{pmatrix}\begin{pmatrix}\theta\\ \dot\theta\\ i_\mathrm{a}\end{pmatrix}\,. \tag{44.48}$$

The discrete state equations sampled from (44.47) and (44.48) with a time interval of $T = 1200\,\mathrm{s}$ are

$$\begin{pmatrix}\theta_{k+1}\\ \dot\theta_{k+1}\\ i_{\mathrm{a},k+1}\end{pmatrix} = \begin{pmatrix}1 & 0.13098 & 0.0051164\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\theta_k\\ \dot\theta_k\\ i_{\mathrm{a},k}\end{pmatrix} + \begin{pmatrix}613.91\\ 0.51164\\ 0.0037955\end{pmatrix}150\,, \tag{44.49}$$

$$Y_k = \begin{pmatrix}0 & 1 & 0\end{pmatrix}\begin{pmatrix}\theta_k\\ \dot\theta_k\\ i_{\mathrm{a},k}\end{pmatrix}\,. \tag{44.50}$$

The following parameters are also used to estimate state in this experiment:

1. Sampling interval $T = 20\,\mathrm{min}$, which is the increment time for each of the steps used in Kalman prediction. In order to compare results for shorter and longer T's, another two estimations with different time intervals, $T = 5\,\mathrm{min}$ and $T = 60\,\mathrm{min}$ were tested.
2. Disturbance $\boldsymbol{W}_k$ has a mean of zero and a variance of 0.1 V [44.22].

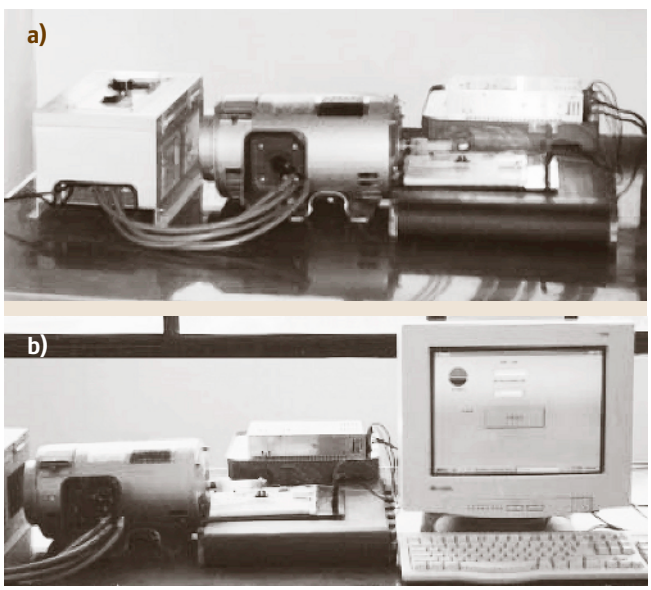


Fig. 44.12 Experiment set-up

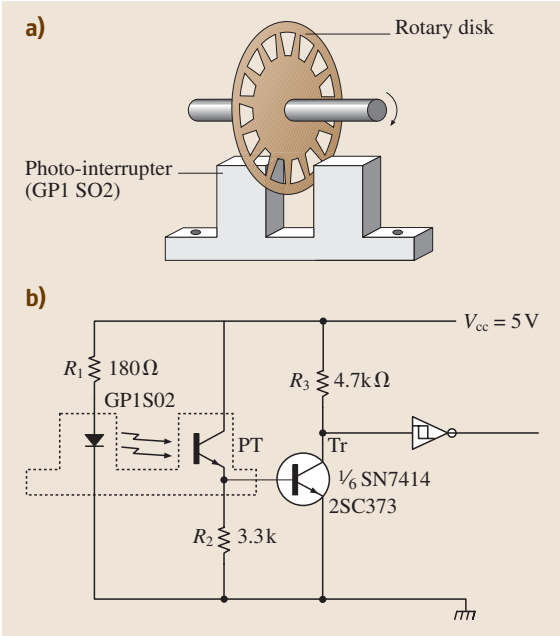


Fig. 44.13 Device and circuit for rotating speed measurement

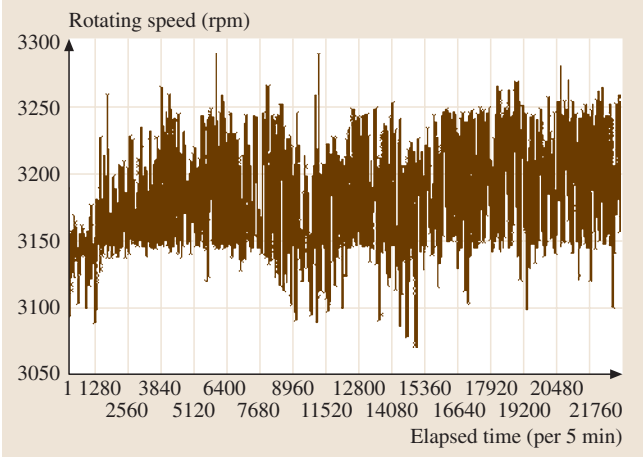


Fig. 44.14 Rotating speed measurements, taken every 5 min

3. The measurement error V_k for $\dot{\theta}$ has a mean of zero and an accuracy of 1% [44.23] of the full-scale measurement.
4. The rated rotation speed of the DC motor is 3180 rpm [44.27], which is therefore the initial value of the state variable $\dot{\theta}$.

Data Acquisition System

The data acquisition system used in this experiment comprises a photo-interrupter circuit, a personal computer (PC), and a RS-232 transmission interface [44.23]. The rotation speed of the DC motor is measured by the photo-interrupter coded GP1S02. The shaft of a rotary disk is connected to the shaft of the DC motor, and the disk is placed between the light-emitting element and the light-receiving element of the photo-interrupter so as to generate pulse-signals while the motor rotates. The device and the circuit are shown in Fig. 44.13.

Pulse-signals are transmitted to the PC through the RS-232 interface, and the PC counts the pulses that are accumulated within 60 seconds in order to derive the rotation speed in rpm (revolution per minute).

44.5.2 Experimental Results

The results of the experiment are presented and discussed in this section.

Measured Data

The rotation speed of the motor was measured and recorded every 5 min day and night for 80 d. Because the experiment lasted for nearly three months, lots of data were accumulated. There are 288 measurements per day and 23 040 data values in total for that period of time. Figure 44.14 shows the results. The data were fed into the estimator, as depicted in Fig. 44.14, in order to estimate the one-step-ahead state variables. The measured data and the resulting estimates for $T = 20$ min (every fourth measurement is used) are shown in Fig. 44.15. The data would be difficult to interpret if all 23 040 of the data values were shown in one chart. To avoid this and therefore to present the results more clearly, the unit used on the time axis of Fig. 44.15 is set to be 24 h (one point per day).

Error in the Estimate

The error in the estimate (in percent) is defined as

$$E_r\% = \frac{\hat{\dot{\theta}}_{k+1/k} - \dot{\theta}_{k+1}}{\dot{\theta}_{k+1}} \times 100\% \,. \qquad (44.51)$$

E_r represents the difference between the predicted value and the actual value. Figure 44.16 shows the results obtained from (44.51) using the data in Fig. 44.15. Reading from Fig. 44.16, the maximum $E_r\%$ is less than 3%.

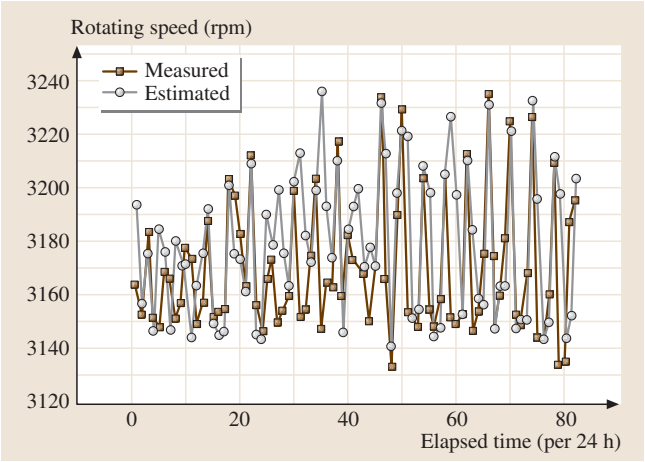


Fig. 44.15 Measured and estimated motor rotating speeds

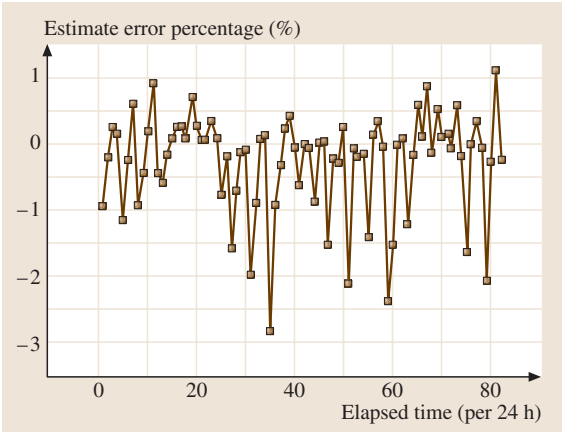


Fig. 44.16 Percent errors in estimates

Mean Value and Variance of the Accuracy of the Estimate

Let

$$X_i = 1 - E_r\%_i \quad i = 1, 2, \ldots, 23\,040 \tag{44.52}$$

be the individual accuracy of each estimate, and

$$\mu = \frac{\sum_{i=1}^{23\,040} X_i}{23040}\,, \tag{44.53}$$

$$\sigma^2 = \frac{\sum_{i=1}^{23\,040} (X_i - \mu)^2}{23\,040} \tag{44.54}$$

be the mean value and the variance [44.25] of the accuracy for the 23 040 samples, respectively. The resulting mean values, variances, and standard deviations (σ) for the estimated accuracy for $T = 5, 20, 60$ min, obtained using (44.52), (44.53), and (44.54), are summarized in Table 44.1.

Table 44.1 Mean values, standard deviations, and variances for different T

T min	μ(%)	σ(%)	σ^2(%)
5	99.74656	0.402957	0.162374
20	99.74060	0.471612	0.222418
60	99.72771	0.652469	0.425716

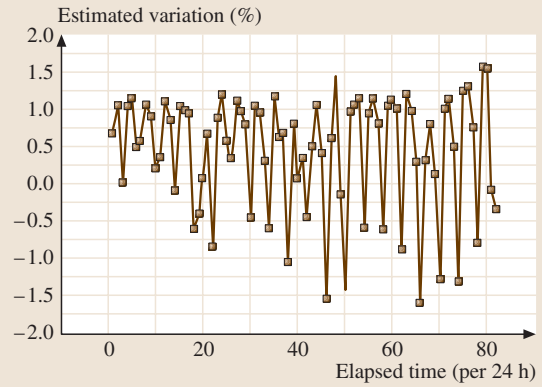


Fig. 44.17 Percent variation in the estimated rotating speed

Variation in Rotation Speed

The variation in estimated rotation speed for the DC motor (in percent) is defined using

$$V\% = \frac{\hat{\theta}_{k+1/k} - 3180}{3180} \times 100\% \,. \tag{44.55}$$

$V\%$ represents the percentage variation in the estimated rotation speed from the rated value 3180 rpm; in other words the abnormality in the motor's performance. It is used to judge whether the motor is going to fail or not. Since the MTBF of the motor is about 100 000 h [44.21], the rotation speed of the motor in this experiment varied by less than 2% of the rated value over the experiment time period. The percentage variation in the estimated rotation speed of the DC motor is shown in Fig. 44.17.

44.5.3 Notes About the Experiment

1. The mean accuracies of the estimates for $T = 5$, 20, and 60 min were all higher than 99.7%, which infers that the one-step-ahead state variable can be accurately predicted using the proposed method.

2. A threshold is a value used to judge whether or not an equipment failure occurs. It is defined as the measurement value that is taken just prior to or at the time of failure [44.28]. For failure prediction, the threshold for the motor should be determined by the user of the motor according to requirements of the specific situation. Once the estimated value reaches the threshold, a failure is predicted.
3. The disturbance amplitude should encompass all possible uncertainties for the motor and the environment.
4. Since the prediction is used for the purposes of PM, the prediction time should be long enough that PM can be performed before failure occurs.
5. The method proposed in this study is exemplified by a motor system, which is treated as a component. The procedure can be performed on a multicomponent system if state equations can be constructed for the components as a whole. It is feasible to perform the procedure on either the multicomponent system or each of the components. For a complicated or large system, the proposed method can be performed on those elements present in minimum cut sets as constructed by fault tree analysis or the Petri net model for failure [44.26].

44.6 Conclusions

Knowing when and where a system needs maintenance and economizing capital investment are two of the major issues associated with maintenance. The proposed scheme optimizes maintenance in the following aspects:

1. Before a system failure occurs, the scheme is able to indicate where and when the failure is going to be.
2. It makes the health of the system and its historical record clear at a glance.
3. Scheduled maintenance is performed according to a statistical average, which still retains an unavoidable risk that the system may fail before the criteria are exceeded, so a failure may occur unexpectedly. On the other hand, the actual duty cycles for a certain part or module may be longer than the average, so it is a waste of investment if they are replaced during scheduled maintenance. The condition-based scheme avoids those drawbacks.

Failure prediction simulation and experiment for PM, performed via state estimation using Kalman filtering, was described in this chapter. In contrast to previous works, this study uses Kalman filtering instead of parameter trends to predict the time of failure and to determine the PM execution time. The resulting prediction errors are acceptable for not only one-step-ahead prediction but also two-steps-ahead prediction. To simulate the aging failure mode, a state variable – the rotation speed – is monitored in the simulation. More measured variables mean that more complicated failure modes can be simulated. Moreover, the chapter also described an experiment on a DC motor used for state estimation via predictive maintenance using the Kalman filter. The resulting prediction errors for one-step-ahead prediction were acceptable. The shorter the increment time for Kalman prediction, the higher the prediction accuracy. Considerations used to determine the PM lead-time and the increment time required for prediction contradict each other. How to work out a compromise and end up with an optimal value is an important issue. By incorporating fault tree analysis or a Petri net model for failure, the proposed method can be performed on only those elements of a complicated or large system that fall into minimum-cut sets instead of all of the elements of the system. Therefore, failure can be prevented in enough time to promote reliability if state estimation using the Kalman filter is applied to predictive maintenance.

References

44.1 E. A. Elsayed: *Reliability Engineering* (Addison Wesley, Reading 1996)

44.2 J. D. Patton, Jr.: *Preventive Maintenance* (Instrument Soc. Am., Research Triangle Park 1983)

44.3 IEV: *International Electrotechnical Vocabulary (IEV); Chapt. 191: Dependability and quality of service* (International Electrotechnical Commission, Geneva 1990)

44.4 M. Rausand, K. Oien: The basic concept of failure analysis, Reliab. Eng. Syst. Safe. **53**, 73–83 (1996)

44.5 S. S. Rao: *Reliability-Based Design* (McGraw-Hill, New York 1992)

44.6 O. T. Ogunyemi, P. I. Nelson: Prediction of Gamma failure times, IEEE Trans. Reliab. **46**(3), 400–405 (1997)

44.7 W. Nelson: Weibull prediction of a future number of failures, Qual. Reliab. Eng. Int. **16**, 23–26 (2000)

44.8 E. A. Rietman, M. Beachy: A study on failure prediction in a plasma reactor, IEEE Trans. Semiconduct. M. **11**(4), 670–680 (1998)

44.9 S. N. Kher, G. M. Bubel: Predicting system-failure risk from unanticipated fiber-breaks in manufacturing, IEEE Trans. Reliab. **47**(2), 126–130 (1998)

44.10 R. Isermann: Process fault detection based on modeling and estimation methods – A survey, Automatica **20**(4), 387–404 (1984)

44.11 J. L. Tylee: On-line failure detection in nuclear power plant instrumentation, IEEE Trans. Automat. Contr. **AC-28**(3), 406–415 (1983)

44.12 M. M. Sidar, B. F. Doolin: On the feasibility of real-time prediction of aircraft carrier motion at sea, IEEE Trans. Automat. Contr. **AC-28**(3), 350–355 (1983)

44.13 R. F. Berg: Estimation and prediction for maneuvering target trajectories, IEEE Trans. Automat. Contr. **AC-28**(3), 294–304 (1983)

44.14 B. D. O. Anderson, J. B. Moore: *Optimal Filtering* (Prentice-Hall, Englewood Cliffs 1979)

44.15 A. S. Willsky: A survey of design methods for failure detection in dynamic systems, Automatica **12**, 601–611 (1976)

44.16 R. G. Brown, P. Y. C. Hwang: *Introduction to Random Signals and Applied Kalman Filtering* (Wiley, New York 1997) pp. 198–224 Sec. 5.3-5.6

44.17 K. Ogata: *Modern Control Engineering* (Prentice-Hall, Englewood Cliffs 1980)

44.18 C. L. Phillips, H. T. Nagle: *Digital Control System Analysis and Design* (Prentice-Hall, Englewood Cliffs 1990)

44.19 K. Ogata: *Discrete-Time Control Systems* (Prentice-Hall, Englewood Cliffs 1987)

44.20 G. F. Franklin, J. D. Powell, A. Emami-Naeini: *Feedback Control of Dynamic Systems* (Addison-Wesley, New York 1994)

44.21 B. S. Dhillon: *Mechanical Reliability: Theory, Models and Applications* (AIAA, Washington DC 1988)

44.22 C. Y. Chiang-Lin: *Parameter estimation and fault diagnosis on the thermal network by applying the extended Kalman filter and the expert system, M.S. Thesis* (National Chiao Tung University, Taiwan 1991)

44.23 J. G. Webster, R. Palla's-Areny: *Sensor and Signal Conditioning* (Wiley, New York 1991)

44.24 D. Fowley, M. Horton: *User's Guide to MATLAB, Version 4* (Prentice-Hall, Englewood Cliffs 1995)

44.25 K. Black: *Business Statistics, Contemporary Decision Making* (West Publ., Los Angeles 1997)

44.26 S. K. Yang, T. S. Liu: Failure analysis for an airbag inflator by Petri nets, Qual. Reliab. Eng. Int. **13**, 139–151 (1997)

44.27 TECO: *DC Motor, Universal Type: User Guide* (Tung-Yuang Electrical Machine Co. (TECO), Taipia 2001)

44.28 S. K. Yang, T. S. Liu: A Petri net approach to early failure detection and isolation for preventive maintenance, Qual. Reliab. Eng. Int. **14**, 319–330 (1998)

45. Statistical Maintenance Modeling for Complex Systems

The first part of this chapter provides a brief introduction to statistical maintenance modeling subject to multiple failure processes. It includes a description of general probabilistic degradation processes.

The second part discusses detailed reliability modeling for degraded systems subject to competing failure processes without maintenance actions. A generalized multi-state degraded-system reliability model with multiple competing failure processes including degradation processes and random shocks is presented. The operating condition of the multi-state system is characterized by a finite number of states. A methodology to generate the system states when multi-failure processes exist is also discussed. The model can be used not only to determine the reliability of the degraded systems in the context of multi-state functions but also to obtain the probabilities of being in a given state of the system.

The third part describes the inspection–maintenance issues and reliability modeling for degraded repairable systems with competing failure processes. A generalized condition-based maintenance model for inspected degraded systems is discussed. An average long-run maintenance cost rate function is derived based on an expression for degradation paths and cumulative shock damage, which are measurable. An inspection sequence is determined based on the minimal maintenance cost rate. Upon inspection, a decision will be made on whether to perform preventive maintenance or not. The optimum preventive maintenance thresholds for degradation processes and inspection sequences are also determined based on a modified Nelder–Mead downhill simplex method.

Finally, the last part is given over to the conclusions and a discussion of future perspectives for degraded-system maintenance modeling.

Technology advances mean that most new products are, on one hand, more reliable, but on the other hand, very difficult to maintain during the product life cycle. Designers have been challenged to find new, effective approaches to evaluate reliability in a timely fashion and to maintain such systems in an optimum way. This chapter presents reliability and maintenance models for degraded systems subject to competing failure processes. The accuracy of reliability estimation through a degradation model cannot be ensured unless the unit-to-unit initial variation and within-unit degradation-rate variation are considered. This chapter also discusses a generalized random-coefficient degradation process and randomized logistic degradation process to model the degradation.

Literature Review

As degradation occurs, system performance changes from perfect functioning to complete failure; the binary assumption used to analyze, model and compute system reliability is relaxed. Using degradation measures to assess reliability has seen some important findings in the literature. *Tomsky* [45.1] investigated two regression models for detecting degradation reliability. *Nelson* [45.2] briefly surveyed the application of accelerated degradation. *Lu* [45.3] introduced a nonlinear mixed-effects model and estimated model parameters in a two-stage way. Recently, multi-state reliability has received considerable attention. *Levitin* [45.4] extended the reliability importance measures for multi-state systems with different measures of performance. When the multi-state nature of a system is addressed, a better understanding of the system reliability behavior is obtained. Our third new development is to build a methodology based on the formulation of degradation in terms of a finite discrete state.

It is well known that the effectiveness of a system depends on both the quality of its design and manufacturing process as well as the proper inspection–maintenance actions to prevent it from failing. Inspection–maintenance issues are considered in the second part of this chapter.

Maintenance has evolved from a simple model that deals with machinery breakdowns, to time-based preventive maintenance, to today's condition-based maintenance. It is of great importance to avoid the failure of a system during its actual operating; especially, when such failures are dangerous and costly. Time-based and condition-based maintenance are the two major approaches for maintenance. Condition-based maintenance is often profitable since it can be used to avoid failure occurrence at the lowest cost and to improve the availability and reliability of complex systems. This chapter examines the problem of developing a mathematical maintenance cost model to determine both the optimal inspection interval time and preventive maintenance threshold of degraded systems with competing failure processes subject to a condition-based maintenance policy.

Pham et al. [45.5] presented a Markov model for predicting the reliability of k-out-of-n systems in which components are subject to multi-stage degradation as well as catastrophic failures. Due to the aging effect, the failure rate of the component will increase. They considered the state-dependent transition rates for the degradation process.

Pham et al. [45.6] derived models for predicting the availability and mean lifetime of multistage degraded systems with partial repairs.

In some production systems failures are not possible to detect but can only be determined by inspection [45.7]. Several authors [45.8–22] have proposed various inspection policies and models for systems with a degradation process. *Grall* et al. [45.8] studied a system subject to a random deterioration process. They developed a model based on a stationary process to determine both the preventive maintenance threshold and inspection dates that minimized the average long-run cost rate. *Chelbi* and *Ait-Kadi* [45.10] addressed optimal inspection strategies for deteriorating equipment subject to preventive and corrective maintenance.

Klutke and *Yang* [45.11] studied the availability of maintained systems subject to both the effects of the degradation and random shocks. They considered the degradation process as a deterministic function of time t and that shocks occurred according to a Poisson process in which the shock magnitudes are independent and identically distributed (iid) random variables. *Pham* and *Xie* [45.13] developed a generalized surveillance model consisting of dual, mutually dependent stochastic processes for surveillance systems. Their model can be used to better understand both the inspection process, the repair unit itself, and to provide information that can be used to assist inspectors in scheduling and prioritizing their future inspections.

The choice of the inspection schedule and preventive maintenance threshold(s) obviously has an important influence on the economic performance of the maintenance policy. The inspection dates and the preventive maintenance threshold(s) are two main decision variables. However, in industrial applications of condition-based maintenance, the preventive maintenance threshold is usually decided based upon the recommendation made by the maintenance people and the inspection schedule often appears to be set by little more than a rule of thumb. Because of the lack of appropriate modeling support, the preventive maintenance threshold is likely to be set conservatively and the inspection schedule may be performed more than is perhaps necessary. The need for a maintenance model with cost consideration is obvious in this case.

The chapter is organized as follows. The basic concepts and a review of maintenance, as well as a brief description of probabilistic processes for the modeling of degradation and random shocks, is discussed in Sect. 45.1. A general reliability model for degraded nonrepairable systems subject to multiple competing

processes is discussed in Sect. 45.2. The inspection–maintenance modeling issues and detection policies for degraded repairable systems considering multiple competing processes are described in Sect. 45.3. Several numerical examples are given in Sects. 45.2 and 45.3. Finally in Sect. 45.4, several future research perspectives and conclusions are briefly discussed.

Acronyms

CM	Corrective maintenance
PM	Preventive maintenance
rv	Random variable

Notation

C_c	Cost per CM action
C_p	Cost per PM action
C_m	Loss per unit idle time
C_i	Cost per inspection
$Y(t)$	Degradation process
$Y_1(t)$	Degradation process 1
$Y_2(t)$	Degradation process 2
$D(t)$	Cumulative shock damage value up to time t
G_1	Critical threshold value for degradation process 1
G_2	Critical threshold value for degradation process 2
S	Critical value for shock damage
L_1	PM critical threshold value for degradation process 1
L_2	PM critical threshold value for degradation process 2
$C(t)$	Cumulative maintenance cost up to time t
$E[C_1]$	Average total maintenance cost during a cycle
$E[W_1]$	Mean cycle length
$E[N_I]$	Mean number of inspections during a cycle
$E[\xi]$	Mean idle time during a cycle
$\{I_i\}_{i\in N}$	Inspection sequence
$\{U_i\}_{i\in N}$	Inter-inspection sequence
$\{W_i\}_{i\in N}$	Renewal times
T	Time to failure
P_{i+1}	Probability that there are a total of $(i+1)$ inspections in a renewal cycle
P_p	Probability that a renewal cycle ends by a PM action
P_c	Probability that a renewal cycle ends by a CM action
$EC(L_1, L_2, I_1)$	Expected long-run cost rate function
Ω_U	$\{M, \ldots, 1, 0, F\}$ a system state space where state M is a perfect (good) state; state 0 is a degraded failure state; states $M-1 \ldots l$ are intermediate degradation states; state F is the catastrophic failure state
Ω	$\{M, \ldots, 1, 0\}$, a system degradation state space without catastrophic failure
Ω_i	$\{M_i, \ldots, 1_i, 0_i\}$, a state space corresponding to the degradation processes of the i-th state, where 0_i is a degraded failure state due to the i-th degradation process, and M_i is a good state of the degradation process i, $i = 1, 2$
R	$\Omega_1 \times \Omega_2$, the Cartesian product of Ω_1 and Ω_2
R_i	i-th equivalence class, $i = 0, 1, \ldots, M$

45.1 General Probabilistic Processes Description

We consider three random processes. The first two are used to model degradation, while the third is a compound Poisson process used for modeling random shocks:

1. $Y(t) = A + Bg(t)$ is called a random-coefficient degradation path, where $A > 0$and $B > 0$ are independent random variables and $g(t)$ is an increasing time-dependent function. The random variable A measures the initial value of degradation due to a different manufacturer, the manufacturing quality control of new items, variable deterioration during storage until the item is put into service, and so forth [45.14]. Therefore, the initial degradation value A is a random variable. The variable B is the degradation rate $(B > 0)$ and represents the variations among the population; $g(t)$ is an increasing function.
2. $Y(t) = \frac{W e^{Bt}}{A+e^{Bt}}$ is called a randomized logistic degradation path function where A and B are independent non-negative random variables, and W is a constant. The random variable A represents the initial threshold level of degradation and B describes the rate at which degradation accumulates. It should be noted that $Y(t) = \frac{W e^{Bt}}{A+e^{Bt}}$ is an S-shaped curve and describes the degradation process well. It matches the

path of the cumulative degradation of many systems in practice. The S-shaped curve reflects an initial run-in period of low usage, followed by a period of steady rate of usage, and finally ending with an increasing rate of use due to the aging of the system. We establish the relationship between the two random variables A and B via some rearrangements as follows:

$$W\frac{e^{Bt}}{A+e^{Bt}} < H \Rightarrow B < \frac{1}{t}\ln\frac{u_1 A}{1-u_1}\,, \qquad (45.1)$$

where $u_1 = \frac{H}{W}$.

3. Let $D(t) = \sum_{i=0}^{N(t)} X_i$ represent a sequence of random shocks in which each shock causes independent damage X_i to the whole system where the X_i are iid with a probability distribution function (pdf) of $f_X(x)$, and a cumulative distribution function (cdf) of $F_X(x)$; $\{N(t), t \geq 0\}$ is a Poisson process with parameter $\lambda > 0$ that is independent of the sequence $\{X_i\}$; $F_X^{(k)}(x)$ denotes the k-th convolution.
The stochastic process $D(t) = \sum_{i=0}^{N(t)} X_i$ is called a compound Poisson process where $N(t)$ is the number of shocks that have occurred up to time t, X_i is the damage caused by the i-th shock and $D(t)$ is the cumulative damage up to time t.

45.2 Nonrepairable Degraded Systems Reliability Modeling

This section addresses reliability models for nonrepairable degraded systems. First, we discuss a model for systems subject to two competing processes. Then, we present a generalized situation where systems are subjected to three competing processes.

45.2.1 Degraded Systems Subject to Two Competing Processes

Model description

The modeling assumptions are as follows:

1. Each system has a state space $\Omega_U = \{M, \ldots, 1, 0, F\}$.
2. The system fails either due to degradation $(Y(t) > G)$ or catastrophic failure $\left[D(t) = \sum_{i=1}^{N_2(t)} X_i > S\right]$. The system may either go from state i to the next degraded state $i-1$ or directly to the catastrophic failure state F, $i = M, \ldots, 1$.
3. No repair or maintenance is performed on the system.
4. Since $Y(t)$ describes the total damage up to time t, it is natural to assume that it is nondecreasing.
5. The two processes $Y(t)$ and $D(t)$ are independent.
6. At time $t = 0$, the system is in state M.

We consider a degradable system suited at a random environment where degradation and random shocks can contribute to an effect of the life of a system. In this section, we discuss the case where systems are subject to two failure processes, called a continuous and increasing degradation process $Y(t)$, and the a andom shock process $D(t)$. Whichever process occurs first causes the system to failure.

Figure 45.1 illustrates the system flow diagram of the two competing failure processes. In Fig. 45.1, we use either of the random processes described in Sect. 45.1 to represent a degradation process where random shocks are represented by a stationary and independent increment process. Then, we discuss a method to formulate these two processes from a multi-state standing point. That is, suppose that the operating conditions of the system at any point in time could be classified into one of a finite number of the states, say $\Omega_U = \{M, \ldots, 1, 0, F\}$. We view the degradation process in terms of a finite number of states. For example, when the value of the degradation process $Y(t)$ falls into a predefined interval then its corresponding state will be determined. Let us define as follows:

$[0, W_M], \ldots, (W_2, W_1]$ are the intervals associated with the degradation process where $W_M < W_{M-1} < \ldots < W_2 < W_1$. A one-to-one relationship between the element of $\Omega = \{M, \ldots, 1, 0\}$ and its corresponding interval is set up as follows:

$$\begin{aligned}
\text{when } Y(t) &\in [0, W_M] \Rightarrow \text{in state } M\,,\\
\text{when } Y(t) &\in \left(W_M, W_{M-1}\right] \Rightarrow \text{in state } M-1\,,\\
&\vdots\\
\text{when } Y(t) &\in \left(W_i, W_{i-1}\right] \Rightarrow \text{in state } i\,,\\
\text{when } Y(t) &\in \left(W_2, W_1\right] \Rightarrow \text{in state } 1\,,\\
\text{when } Y(t) &> W_1 \Rightarrow \text{in state } 0\,.
\end{aligned}$$

Reliability Evaluation

The most general situation is to allow each degradation process to be described by a number of different discrete

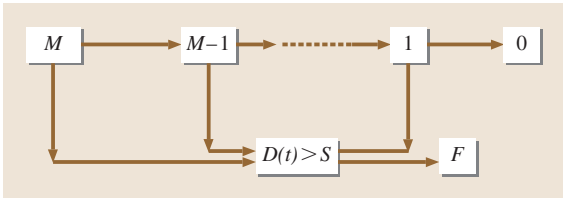


Fig. 45.1 The flow diagram of a system subjected to two competing failure processes

states. We now define the probability in each state. Let $P_i(t)$ be the probability that the value of $Y(t)$ will fall within a predefined interval corresponding to state i with $D(t) \le S$. From state i, the system will make a direct transition to state $i-1$ due to gradual degradation, or to state F due to a random shock.

The reliability function is defined as:

$$\begin{aligned} R_M(t) &= P\,(\text{state} \ge 1) \\ &= \sum_{i=1}^{M} P_i(t) \\ &= P[Y(t) \le G,\, D(t) \le S]\,, \end{aligned} \tag{45.2}$$

where $P_i(t)$ is the probability of being in state i.

Suppose a system fails if the degradation process crosses some threshold, say G; or the shock damage process crosses some threshold, say S; T is defined as:

$$T = \inf\,[t > 0 : Y(t) > G \text{ or } D(t) > S] \tag{45.3}$$

The mean time to failure is expressed as:

$$\begin{aligned} E[T] &= \int_0^\infty P[T > t]\,\mathrm{d}t \\ &= \int_0^\infty P[Y(t) \le G,\, D(t) \le S]\,\mathrm{d}t \\ &= \int_0^\infty P[Y(t) \le G] \sum_{j=0}^{\infty} \frac{(\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t}}{j!} F_X^{(j)}(S)\,\mathrm{d}t \end{aligned}$$

or, equivalently, that

$$E[T] = \sum_{j=0}^{\infty} \frac{F_X^{(j)}(S)}{j!} \int_0^\infty P[Y(t) \le G](\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t}\,\mathrm{d}t\,. \tag{45.4}$$

The specific expression for $E[T]$ depends on the probability function $P[Y(t) \le G]$. Sometimes, it is hard to find a closed-form solution. In this case, one can use a numerical method to solve the problem in (45.4).

The probability density function of the time to failure, $f_T(t)$ is as follows:

$$\begin{aligned} f_T(t) &= -\frac{\mathrm{d}}{\mathrm{d}t} R(t) \\ &= -\frac{\mathrm{d}}{\mathrm{d}t} \{P\,[Y(t) \le G]\,P\,[D(t) \le S]\} \\ &= -\frac{\mathrm{d}}{\mathrm{d}t} \left\{ P[Y(t) \le G] \sum_{j=0}^{\infty} \frac{(\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t}}{j!} F_X^{(j)}(S) \right\} \\ &= -\sum_{j=0}^{\infty} \frac{F_X^{(j)}(S)}{j!} \frac{\mathrm{d}}{\mathrm{d}t} \left\{ P[Y(t) \le G]\,(\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t} \right\} \end{aligned}$$

Let $F_G(t) = P[Y(t) \le G]$, then $f_G(t) = \frac{\mathrm{d}}{\mathrm{d}t} F_G(t)$.

$$\begin{aligned} f_T(t) = -\sum_{j=1}^{\infty} \frac{F_X^{(j)}(S)}{j!} \Big[& f_G(t)(\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t} \\ & + F_G(t)\, j\lambda_2 (\lambda_2 t)^{j-1}\, \mathrm{e}^{-\lambda_2 t} \\ & - \lambda_2 F_G(t)(\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t} \Big]\,. \end{aligned} \tag{45.5}$$

Reliability models

Model 1:

$$\begin{cases} Y(t) = A + Bg(t) \\ D(t) = \sum_{i=0}^{N_2(t)} X_i \end{cases}$$

$$\begin{cases} \text{case 1}: A \sim \text{normal},\ B \sim \text{normal} \\ \text{case 2}: A \sim U[0, a],\ B \sim \text{Exp}(b) \end{cases}$$

Model 2:

$$\begin{cases} Y(t) = W \frac{\mathrm{e}^{Bt}}{A + \mathrm{e}^{Bt}}, \text{ where } A \sim U[0, a],\ B \sim \text{Exp}(b) \\ D(t) = \sum_{i=0}^{N_2(t)} X_i \end{cases}$$

The two reliability models for the system are depicted in Fig. 45.1. In the following, we will take model 2 as an example to illustrate the results in this section. One can also easily apply it for the model 1.

Assume that the degradation process is described by the function $Y(t) = W \frac{\mathrm{e}^{Bt}}{A + \mathrm{e}^{Bt}}$, where the two random variables A and B are independent, and A follows a uniform distribution with parameter interval [0, a] and B follows an exponential distribution with parameter $\beta > 0$. In short, $A \sim U[0, a]$, $a > 0$ and $B \sim \text{Exp}(\beta)$, $\beta > 0$.

The probability that the system is in state M is as follows:

$$
\begin{aligned}
P_M(t) = P\Bigg[& Y(t) = W\frac{\mathrm{e}^{Bt}}{A+\mathrm{e}^{Bt}} \le W_M\,, \\
& D(t) = \sum_{i=0}^{N_2(t)} X_i \le S\Bigg] \\
= & \left[\int_{\forall A} P\left(B < \frac{1}{t}\ln\frac{u_1 A}{1-u_1} | A = x\right) f_{\mathrm{A}}(x)\,\mathrm{d}x\right] \\
& \times P\left[D(t) = \sum_{i=0}^{N_2(t)} X_{\mathrm{i}} \le S\right] \\
= & \left[1 - \frac{1}{a}\left(\frac{1-u_1}{u_1}\right)^{\frac{\beta}{t}}\left(\frac{t}{t-\beta}\right)\left(a^{1-\frac{\beta}{t}} - 1\right)\right] \\
& \times \mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\frac{(\lambda_2 t)^j}{j!}F_X^{(j)}(S)\,. \qquad (45.6)
\end{aligned}
$$

The probability that the system is in state i is calculated as follows:

$$
\begin{aligned}
P_{\mathrm{i}}(t) = P\Bigg[& W_{\mathrm{i}+1} < W\frac{\mathrm{e}^{Bt}}{A+\mathrm{e}^{Bt}} \le W_i\,, \\
& D(t) = \sum_{i=0}^{N_2(t)} X_{\mathrm{i}} \le S\Bigg] \\
= & \left[\int_0^a P\left(\frac{1}{t}\ln\frac{u_{\mathrm{i}-1}A}{1-u_{i-1}} < B \right.\right. \\
& \left.\left. \le \frac{1}{t}\ln\frac{u_{\mathrm{i}}A}{1-u_{\mathrm{i}}} | A = x\right) f_{\mathrm{A}}(x)\,\mathrm{d}x\right] \\
& \times \mathrm{e}^{-\lambda_2 t}\sum_{j=1}^{\infty}\frac{(\lambda_2 t)^j}{j!}F_X^{(j)}(S) \\
= & \left\{\frac{1}{a}\left(\frac{t}{t-\beta}\right)\left(a^{1-\frac{\beta}{t}}\right)\left[\left(\frac{1-u_{\mathrm{i}}}{u_{\mathrm{i}}}\right)^{\frac{\beta}{t}} \right.\right. \\
& \left.\left. - \left(\frac{1-u_{i-1}}{u_{i-1}}\right)^{\frac{\beta}{t}}\right]\right\} \\
& \times \mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\frac{(\lambda_2 t)^j}{j!}F_X^{(j)}(S)\,, \qquad (45.7)
\end{aligned}
$$

where $\mu_i = \frac{W_i}{W}$, $i = M-1, .., 1$.

Similarly, the probability that the system is in state 0 is as follows:

$$
\begin{aligned}
P_0(t) = P\Bigg[& Y(t) = W\frac{\mathrm{e}^{Bt}}{A+\mathrm{e}^{Bt}} > G\,, \\
& D(t) = \sum_{i=0}^{N_2(t)} X_{\mathrm{i}} \le S\Bigg] \\
= & \left[\frac{1}{a}\left(\frac{1-u_M}{u_M}\right)^{\frac{\beta}{t}}\left(\frac{t}{t-\beta}\right)\left(a^{1-\frac{\beta}{t}}\right)\right]\mathrm{e}^{-\lambda_2 t} \\
& \times \sum_{j=0}^{\infty}\frac{(\lambda_2 t)^j}{j!}F_X^{(j)}(S)\,. \qquad (45.8)
\end{aligned}
$$

The probability for a catastrophic failure state F is given by:

$$
\begin{aligned}
P_{\mathrm{F}}(t) = P\Bigg[& Y(t) = W\frac{\mathrm{e}^{Bt}}{A+\mathrm{e}^{Bt}} \le G\,, \\
& D(t) = \sum_{i=0}^{N_2(t)} X_i > S\Bigg] \\
= & \left[1 - \frac{1}{a}\left(\frac{1-u_1}{u_1}\right)^{\frac{\beta}{t}}\left(\frac{t}{t-\beta}\right)\left(a^{1-\frac{\beta}{t}}\right)\right] \\
& \times \left[1 - \mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\frac{(\lambda_2 t)^j}{j!}F_X^{(j)}(S)\right]\,. \qquad (45.9)
\end{aligned}
$$

The reliability $R_M(t)$ is expressed as:

$$
\begin{aligned}
R_M(t) = & \sum_{k=1}^{M} P_{\mathrm{k}}(t) \\
= & \left[1 - \frac{1}{a}\left(\frac{1-u_M}{u_M a}\right)^{\frac{\beta}{t}}\left(\frac{t}{t-\beta}\right)\left(a^{1-\frac{\beta}{t}}\right)\right] \\
& \times \left[\mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\frac{(\lambda_2 t)^j}{j!}F_X^{(j)}(S)\right]\,. \qquad (45.10)
\end{aligned}
$$

A Numerical Example

Assume that the degradation is modeled as the function $Y(t) = W\frac{\mathrm{e}^{Bt}}{A+\mathrm{e}^{Bt}}$ where $A \sim U[0, 5]$ and $B \sim \mathrm{Exp}(10)$. The critical values for the degradation and the shock damage are $G = 500$ and $S = 200$, respectively. The random shocks are measured by the function $D(t) = \sum_{i=1}^{N_2(t)} X_{\mathrm{i}}$, where $X_{\mathrm{i}} \sim \mathrm{Exp}(0.3)$ and X_{i}s are

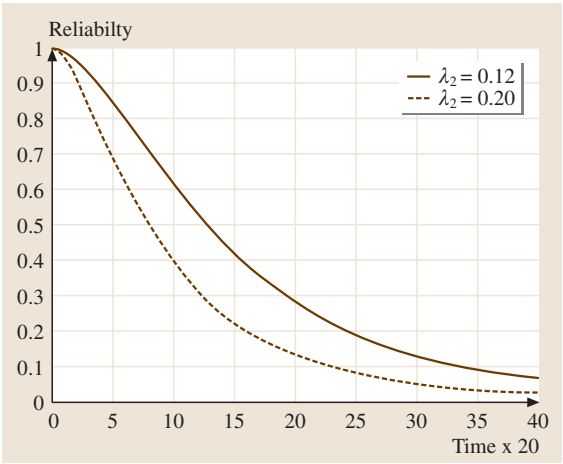


Fig. 45.2 Reliability versus time

iid. Figure 45.2 shows the reliability of the system as a function of time, where the solid line represents $N_2(t)$ with $\lambda_2 = 0.12$ and the dotted line represents $N_2(t)$ with $\lambda_2 = 0.20$.

45.2.2 Systems Subject to Three Competing Processes

System Description

In some applications, the systems are subjected to a variety of governing failure processes. In this section, we consider three independent competing failure processes in which two of them are degradation processes (called degradation process 1, which is measured by the function $Y_1(t)$, and degradation process 2, which is measured by $Y_2(t)$) and the third is a random shock process $D(t)$ [45.15]. Whichever process occurs first causes the system to fail.

Initially, the system is considered to be in its good state (i. e., M_1 and M_2). As time progresses, it can either go to the first degraded state [i. e., $(M-1)_1$ or $(M-1)_2$] upon degradation or can go to a failed state (state F), if subject to random shocks. When a system reaches the first degraded state, it can either stay in that state until the mission time, or it can go to the second degradation state [i. e., $(M-2)_1$ or $(M-2)_2$] upon degradation, or it can go to a failed state (F state) upon random shocks.

The same process will be continued for all stages of degradation except the last degradation, either stage 0_1 or stage 0_2. If the system reaches the last degradation state, it cannot perform its functions satisfactorily and must be treated as a failure (state 0).

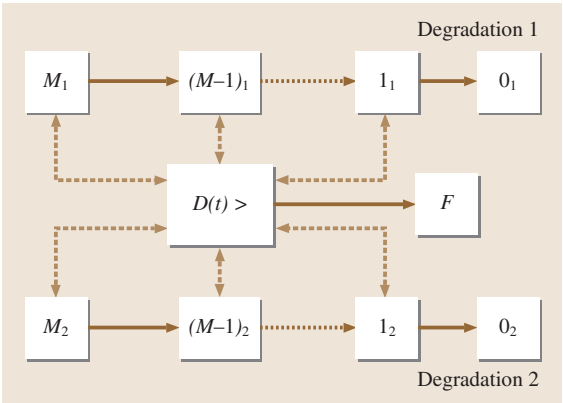


Fig. 45.3 The flow diagram of a system subjected to multiple failure processes [45.15]

Figure 45.3 shows the system flow diagram of the multiple competing transition processes. In Fig. 45.3, the above represents the degradation process 1; the bottom represents the degradation process 2; F represents a catastrophic failure state due to random shocks.

Assumptions.

1. The system consists of $(M+2)$ states where state 0 and state F are both complete failure states. State i is a degradation state, $1 < i < M$.
2. No repair or maintenance is performed on the system.
3. We assume that $Y_i(t)$, $i = 1, 2$ is a nonnegative nondecreasing function at time t, since degradation is an irreversible accumulation of damage.
4. $Y_i(t)$, $i = 1, 2$ and $D(t)$ are statistically independent. The independence assumption implies that the state of one process will have no effect on the state of the others.
5. At time $t = 0$, the system is in state M.
6. The system can fail either due to any of the degradation process when $Y_i(t) > G_i$, $i = 1, 2$ or due to random shocks (in which case it goes to a catastrophic failure state F), i. e. $D(t) = \sum_{i=1}^{N(t)} X_i > S$.
7. The critical threshold value G_i depends upon a function of the states of the degraded systems.

Methodology

In this section, we consider that the degradation paths are modeled by some continuous probabilistic functions. Since the operating condition of the systems is characterized by a finite number of states, let us call the system state space Ω_U. First, we need the discrete continuous

processes. In Step 1 below, we discuss a procedure for forcing two degradation processes to become discrete in order to obtain Ω_1 and Ω_2, which correspond to degradation process 1 and 2, respectively. After we have obtained the degradation process spaces Ω_1 and Ω_2, we present a methodology for how to establish a relationship between the system state space Ω_U and the degradation and random shock state spaces $\{\Omega_1, \Omega_2, F\}$ in Step 2.

Step 1: formulate the degradation processes in terms of discrete state sets. The two-degradation-process case is considered here. The most general situation is to allow each degradation process to be described by a number of different discrete states. The state space denoted by $\Omega_1 = \{M_1, \ldots, 1_1, 0_1\}$ corresponds to degradation process 1 with $M_1 + 1$ states. Similarly, the state space denoted by $\Omega_2 = \{M_2, \ldots, 1_2, 0_2\}$ is associated with degradation process 2, having $M_2 + 1$ states. M_1 and M_2 may or may not be the same, and $M_i < \infty$, $i = 1, 2$.

We view the degradation process from the perspective of a finite number of states. For example, when the value of degradation process 1 $Y_1(t)$ falls into a predefined interval, then its corresponding state will be determined. Let us define as follows:

$[0, W_M], \ldots, (W_2, W_1]$ are the intervals on the degradation 1 curve (Fig. 45.4a) corresponding to state M_1, 0_1, where $W_M < W_{M-1} < \cdots < W_1$ and $[0, A_M], \ldots, (A_2, A_1]$ are intervals associated with the curve for degradation process 2 (Fig. 45.4b) corresponding to state M_2, 0_2, where $A_M < A_{M-1} < \ldots < A_1$.

Mathematically, the relationship between the degradation process states $\Omega_1 = \{M_1, \ldots, 1_1, 0_1\}$, $\Omega_2 = \{M_2, \ldots, 1_2, 0_2\}$, and their corresponding degradation intervals are given as follows:

Degradation process 1

$$0 < Y_1(t) \le W_M, \text{ state } M_1$$
$$W_M < Y_1(t) \le W_{M-1}, \text{ state } (M-1)_1$$
$$\vdots$$
$$W_2 < Y_1(t) \le W_1, \text{ state } 1_1$$
$$G_1 = W_1 < Y_1(t), \text{ state } 0_1$$

Degradation process 2

$$0 < Y_2(t) \le A_M, \text{ state } M_2$$
$$A_M < Y_2(t) \le A_{M-1}, \text{ state } (M-1)_2$$
$$\vdots$$
$$A_2 < Y_2(t) \le A_1, \text{ state } 1_2$$
$$G_2 = A_1 < Y_2(t), \text{ state } 0_2$$

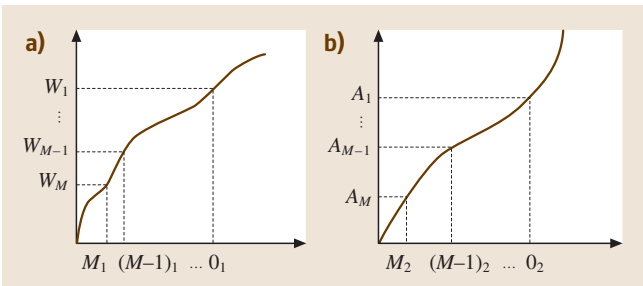


Fig. 45.4a,b The degradation process functions in multi-state terms for: (a) degradation 1, (b) degradation 2

Fig. 45.5 A mapping function

Step 2: generate the system state space. The system state space is defined as $\Omega_U = \{M, \ldots, 1, 0, F\}$, and consists of $M + 2$ states. In this step, we discuss a methodology to develop a function to generate a relationship between the system state space Ω_U and the degradation state spaces $\{\Omega_1, \Omega_2, F\}$. For example, at a given time t, suppose that degradation process 1 is at state $i_1 \in \Omega_1$, and degradation process 2 is at state $j_2 \in \Omega_2$; what is the system state? This question is addressed as follows.

Let us assume that at the current time the system is not in a catastrophic failure state. So state F can be ignored for the time being. Therefore, we can simply look at ways to define a function that has a relationship between Ω and $\{\Omega_1, \Omega_2\}$ instead of Ω_U and $\{\Omega_1, \Omega_2, F\}$.

The operation is described by a mapping function f, which can be written as

$$f : R = \Omega_1 \times \Omega_2 \to \Omega = \{M, .., 1, 0\}$$

where $R = \Omega_1 \times \Omega_2 = \{(i_1, j_2) | i_1 \in \Omega_1, j_2 \in \Omega_2\}$ is a *Cartesian* product as the input space domain, as shown in Fig. 45.5. The matrix H_c given below is an output space consisting of $M + 1$ elements corresponding to

each input-space domain through the function f.

$$H_c = \begin{array}{c} \\ 0_2 \\ 1_2 \\ \vdots \\ M_2 \end{array} \begin{array}{c} \begin{array}{cccc} 0_1 & 1_1 & \cdots & M_1 \end{array} \\ \begin{pmatrix} \times & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & M \end{pmatrix} \end{array} .$$

The top row of this matrix H_c represents the state from degradation process 1. The leftmost column represents the state from degradation process 2. The elements of H_c represent $f(i_1, j_2) = k$ where $i_1 \in \Omega_1$, $j_2 \in \Omega_2$ and $k \in \Omega$. Notice that, in the matrix H_c, all the elements in the first row and first column are zero except that denoted by × because the system will go to a degraded failure state (state 0) when either of the degradations reaches state 0_i, $i = 1, 2$. Besides, some elements in the matrix H_c are also zeros since we define that, when degradation 1 is in some low state $l_1(0_1 < l_1 < M_1)$ and degradation 2 is also in some low state $l_2(0_2 < l_2 < M_2)$, we consider it a degradation failure. It is also observed that $f(M_1, M_2) = M$, because initially the system is in a brand-new state (perfect state M).

As we mentioned above, the first element in H_c is marked by ×, which means it does not exist. The reason is presented as follows. We define the time to failure as

$$T = \inf [t : Y_1(t) > G_1, Y_2(t) > G_2 \text{ or } D(t) > S] . \tag{45.11}$$

It should be noted that all three processes are competing against each other for the life of a system. However, only one of the three processes (whichever occurs first when its corresponding critical threshold value is exceeded) causing the system to fail. Hence, the following events will not happen:

$$P[Y_1(t) > G_1, Y_2(t) > G_2, D(t) \le S] = 0 ,$$
$$P[Y_1(t) > G_1, Y_2(t) > G_2, D(t) > S] = 0 ,$$
$$P[Y_1(t) > G_1, Y_2(t) < G_2, D(t) > S] = 0 ,$$

and

$$P[Y_1(t) < G_1, Y_2(t) > G_2, D(t) > S] = 0 .$$

Because $f(0_1, 0_2) = P[Y_1(t) > G_1, Y_2(t) > G_2, D(t) \le S]$, so the combination of $f(0_1, 0_2)$ does not exist.

The function $f : R = \Omega_1 \times \Omega_2 \to \Omega = \{M, .., 1, 0\}$ is defined with following requirements:

1. $f(0_1, b) = f(a, 0_2) = 0$, where $b \in \Omega_2, a \in \Omega_1$
   $f(M_1, M_2) = M$

Fig. 45.6 A representation of a system state-generating box

2. f is monotonic and nondecreasing in each argument.

   For instance,

   $f(a_1, b_2) \ge f(l_1, b_2)$ if $a_1 \ge l_1$,
   $f(a_1, b_2) \ge f(a_1, l_2)$ if $b_2 \ge l_2$.

Figure 45.6 demonstrates the system's state-generating box. There are two inputs i_1 and j_2 and an output k. The inside mapping mechanism is performed by the function f. At time t, suppose that degradation 1 is at state i_1 and degradation 2 is at state j_2; i_1 and j_2 are inputs. Via matrix H_c, the system state k is then generated as output.

In the matrix H_c different state-combination inputs can generate the same results for the system state. To explain this, we need the following definition of the equivalence class.

Definition 45.1

The i-th equivalence class, R_i, is defined as follows:

$$R_i = [(k_1, j_2) \text{ where } k_1 \in \Omega_1, j_2 \in \Omega_2 | f(k_1, j_2) = i] ,$$
$$i = 0, 1, \ldots, M , \tag{45.12}$$

R_i represents all possible state combinations that generate the system state i; $R_0, \ldots, R_M$ are disjointed sets that partition R into $(M+1)$ equivalence classes, so that

$$R = \bigcup_{i=0}^{M} R_i .$$

45.2.3 Reliability Evaluation

In this section, the probability density functions and the system mean time to failure are derived based on the state probabilities given in Sect. 45.2.1. Now, we derive the probability of being in each state. Initially, the system is in a brand-new state; i. e., in state $M = f(R_M)$. The probability for state M is given by

$$P_t(M) = P_t[f(R_M)] . \tag{45.13}$$

As defined previously, R_i represents all possible state combinations generating the system state i. The probability of being in state i is the union of all the elements

in R_i

$$P_t(i) = P[f(R_i)] \,. \tag{45.14}$$

The probability for a catastrophic failure state F is given by

$$P_t(F) = P[Y_1(t) \le G_1, Y_2(t) \le G_2, D(t) > S] \,. \tag{45.15}$$

The reliability $R(t)$ can be calculated as follows:

$$\begin{aligned} R(t) &= P(\text{system state } \ge 1) \\ &= \bigcup_{i=1}^{M} P[f(R_i)] \\ &= \sum_{i=1}^{M} P_t(i) \,, \end{aligned} \tag{45.16}$$

where $P_t(i)$ is the probability of being in state i.

The mean time to failure is expressed as [45.15]:

$$\begin{aligned} E[T] &= \int_0^{\infty} P(T > t)\,\mathrm{d}t \\ &= \int_0^{\infty} P[Y_1(t) \le G_1] P[Y_2(t) \le G_2] \\ &\quad \times \sum_{j=0}^{\infty} \frac{(\lambda_2 t)^j \, \mathrm{e}^{-\lambda_2 t}}{j!} F_X^{(j)}(S) \end{aligned}$$

or, equivalently, that

$$\begin{aligned} E[T] = \sum_{j=0}^{\infty} \frac{F_X^{(j)}(S)}{j!} \int_0^{\infty} & P[Y_1(t) \le G_1] \\ & \times P[Y_2(t) \le G_2] (\lambda_2 t)^j \, \mathrm{e}^{-\lambda_2 t} \, \mathrm{d}t \,. \end{aligned} \tag{45.17}$$

The result in (45.17) obviously would depend on the expression $P[Y_1(t) \le G_1] P[Y_2(t) \le G_2]$. The probability density function of time to failure, $f_T(t)$ is therefore as follows:

$$\begin{aligned} f_T(t) &= -\frac{\mathrm{d}}{\mathrm{d}t} [P(T > t)] \\ &= -\frac{\mathrm{d}}{\mathrm{d}t} \left\{ P[Y_1(t \le G_1)] P[Y_2(t) \le G_2] \right. \\ &\quad \left. \times \sum_{j=0}^{\infty} \frac{(\lambda_2 t)^j \, \mathrm{e}^{-\lambda_2 t}}{j!} F_X^{(j)}(S) \right\} \,. \end{aligned} \tag{45.18}$$

45.2.4 Numerical Examples

This example aims to illustrate the results discussed in the previous sections. Consider a system subjected to two degradation processes and random shocks.

Assume that degradation process 1 is described by the function $Y_1(t) = A + Bg(t)$, where the random variables A and B are independent and both follow normal distributions, with mean 90 and variance 2.5, and mean 78 and variance 6, respectively. In short, $A \sim N(90, 2.5)$ and $B \sim N(78, 6)$. The degradation function is assumed to be $g(t) = t^3$. Also $G_1 = 2500$, $W_3 = 1500$, $W_2 = 2000$, and $W_1 = 2500$.

Assume that degradation process 2 is described by $Y_2(t) = W \frac{\mathrm{e}^{BBt}}{AA + \mathrm{e}^{BBt}}$, where the random variables AA and BB are independent and follow uniform distributions with parameter interval [0,100] and an exponential distribution with parameter 0.1, respectively. In other words, $AA \sim U[0, 100]$ and $BB \sim \mathrm{Exp}(0.01)$. Also $G_2 = 5000$, $A_2 = 2600$, $A_1 = 5000$, and $W = 7000$. Assume that the random shock is represented by $D(t) = \sum_{i=0}^{N(t)} X_i$ with critical value $S = 200$, where $X_i \sim \mathrm{Exp}(0.1)$ and the X_i are iid.

Assume that the states associated with degradation process 1 and degradation 2 are, respectively, $\Omega_1 = \{3_1, 2_1, 1_1, 0_1\}$ and $\Omega_2 = \{2_2, 1_2, 0_2\}$. We define the system state space as $\Omega_\mathrm{U} = \{3, 2, 1, 0, F\}$ and the matrix H_c is given as

$$H_\mathrm{c} = \begin{array}{c} \\ 0_2 \\ 1_2 \\ 2_2 \end{array} \begin{array}{c} \begin{array}{cccc} 0_1 & 1_1 & 2_1 & 3_1 \end{array} \\ \begin{pmatrix} \times & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{pmatrix} \end{array} .$$

Then we obtain

$$\begin{aligned} R = \{&(0_1, 1_2), (0_1, 2_2), (1_1, 0_2), (2_1, 0_2), (3_1, 0_2), \\ &(1_1, 1_2), (2_1, 1_2), (3_1, 1_2), (1_1, 2_2), (2_1, 2_2), \\ &(3_1, 2_2)\} \end{aligned}$$

The equivalence classes can be listed as follows:

$$\begin{aligned} R_0 &= \{(0_1, 1_2), (0_1, 2_2), (1_1, 0_2), (2_1, 0_2), (3_1, 0_2), \\ &\quad (1_1, 1_2)\} \,, \\ R_1 &= \{(1_1, 2_2)\} \,, \\ R_2 &= \{(2_1, 1_2), (2_1, 2_2)\} \,, \\ R_3 &= \{(3_1, 1_2), (3_1, 2_2)\} \,, \\ R &= \sum_{i=0}^{3} R_i \,. \end{aligned}$$

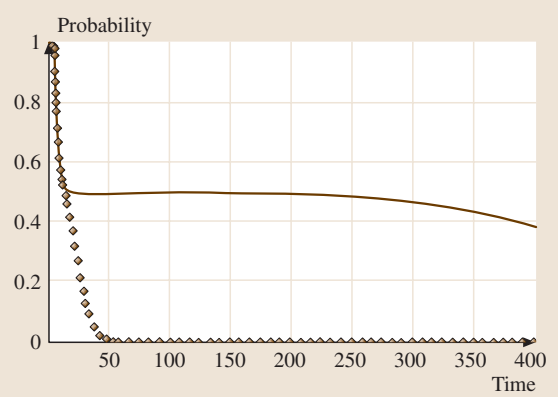


Fig. 45.7 Probability plot for state 3 versus time

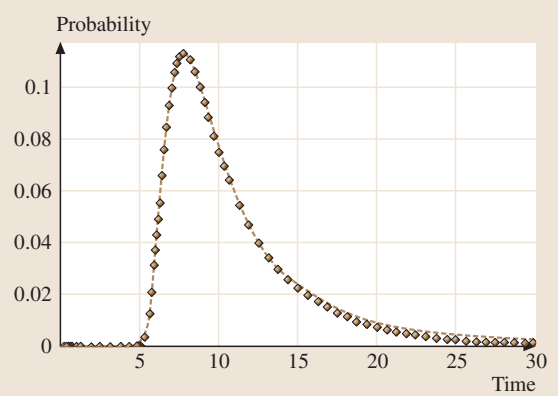


Fig. 45.8 Probability plot for state 2 versus time

According to this expression for H_c, the probability of the system being in state 3 is the sum of the probability $f(3_1, 2_2)$ and of the probability $f(3_1, 1_2)$. That sum is calculated as

$$P_t(3) = P_t\,[f(R_3)] = \Phi\left(\frac{1500-(90+78t)}{\sqrt{2.5+6t^6}}\right)\left[1-\frac{1}{100}\,(0.4)^{\frac{0.01}{t}} \times\left(\frac{t}{t-0.01}\right)\left(0.01^{1-\frac{0.01}{t}}\right)\right] \times \mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\left(\frac{\lambda_2 t}{j!}\right)F_X^{(j)}(200)\,. \quad (45.19)$$

Figure 45.7 shows the probability for the system to be in state 3 as a function of time t, where the solid line represents the compound Poisson process $D(t)=\sum_{i=0}^{N(t)} X_i$ with rate $\lambda = 0.04$ and the dotted line represents the compound Poisson process with rate $\lambda = 0.8$. In Fig. 45.7 we observe that, as t progresses to 50, the probability that the system is in state 3 quickly approaches 0 when the rate is given as $\lambda = 0.8$, and is stable with $\lambda = 0.04$.

Because $R_2 = \{(2_1, 1_2), (2_1, 2_2)\}$, the probability of being in state 2 is given by

$$P_t(2) = P_t[f(2_1, 1_2)] + P_t[f(2_1, 2_2)] = (UV)\,\mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\left(\frac{\lambda_2 t}{j!}\right)F_X^{(j)}(200)\,, \quad (45.20)$$

where

$$U = \Phi\left(\frac{2000-(90+78t)}{\sqrt{2.5+6t^6}}\right) - \Phi\left(\frac{1500-(90+78t)}{\sqrt{2.5+6t^6}}\right),$$

and

$$V = 1-\frac{1}{100}\left(\frac{t}{t-0.01}\right)(0.4)^{\frac{0.01}{t}} \times\left(\frac{t}{t-0.01}\right)(0.01)^{1-\frac{0.01}{t}}\,.$$

Figure 45.8 shows the probability of being in state 2 as a function of time t, where the solid line represents the compound Poisson process $D(t)=\sum_{i=0}^{N(t)} X_i$ with rate $\lambda = 0.04$, and the dotted line represents the compound Poisson process with rate $\lambda = 0.8$. In Fig. 45.8, we observe that, before the time t progresses to 5, the probability of being in state 2 stays close to zero for both rates $\lambda = 0.8$ and $\lambda = 0.04$. It should be noted that the two curves are almost the same for the different values of the rate $\lambda = 0.8$ and $\lambda = 0.04$.

Similarly, the probability of being in state 1 is calculated as:

$$P_t(1) = P_t[f(1_1, 2_2)] = E_1 E_2\,\mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\left(\frac{\lambda_2 t}{j!}\right)F_X^{(j)}(200)\,,$$

where $E_1 = \Phi\left(\frac{2500-(90+78t)}{\sqrt{2.5+6t^6}}\right) - \Phi\left(\frac{2000-(90+78t)}{\sqrt{2.5+6t^6}}\right)$,

$$E_2 = 1-\frac{1}{100}\left(\frac{t}{t-0.01}\right)\left(\frac{22}{13}\right)^{\frac{0.01}{t}}(0.01)^{1-\frac{0.01}{t}}\,. \quad (45.21)$$

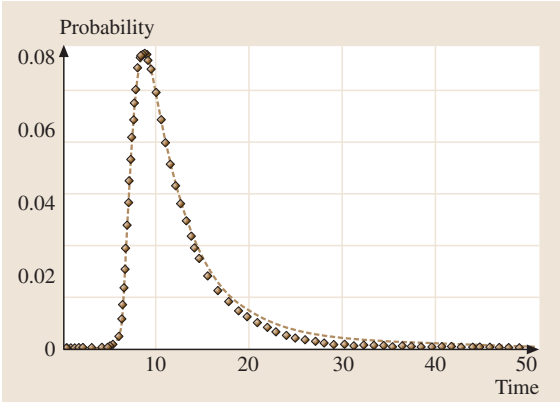


Fig. 45.9 Probability plot for state 1 versus time

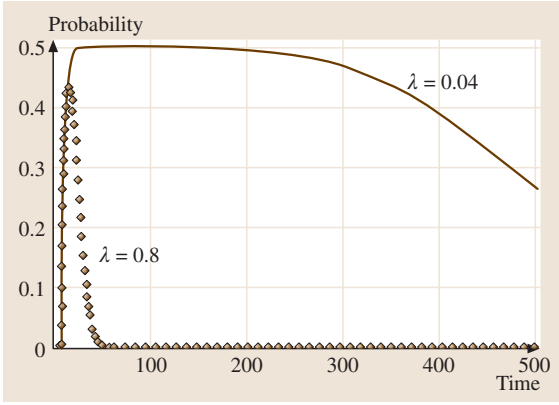


Fig. 45.10 Probability plot for state 0 versus time

Figure 45.9 shows the probability of being in state 1 versus time t, where the solid line represents the compound Poisson process $D(t) = \sum_{i=0}^{N(t)} X_i$ with rate $\lambda = 0.04$, and the dotted line represents the compound Poisson process with rate $\lambda = 0.8$. In Fig. 45.9, we observe that, before the time t progresses to 15, the probability of being in state 1 for both rates $\lambda = 0.8$ and $\lambda = 0.04$ are about the same.

We can also easily obtain the probability of being in state 0 as follows:

$$\begin{aligned} P_t(0) &= P\big[f(0_1, 1_2) + f(0_1, 2_2) + f(1_1, 0_2) \\ &\quad + f(2_1, 0_2) + f(3_1, 0_2) + f(1_1, 1_2)\big] \\ &= (X_1Y_1 + X_2Y_2 + X_3Y_3)\,\mathrm{e}^{-\lambda_2 t} \\ &\quad \times \sum_{j=0}^{\infty} \left(\frac{\lambda_2 t}{j!}\right) F_X^{(j)}(200)\,, \end{aligned}$$

where $X_1 = 1 - \Phi\left(\dfrac{2500-(90+78t)}{\sqrt{2.5+6t^6}}\right)$,

$$X_2 = \Phi\left(\frac{2500-(90+78t)}{\sqrt{25+6t^2}}\right),$$

$$Y_1 = 1 - \frac{1}{100}\,(0.4)^{\frac{0.01}{t}} \left(\frac{t}{t-0.01}\right) \times \left(0.01^{1-\frac{0.01}{t}}\right),$$

$$Y_2 = \frac{1}{100}\,(0.4)^{\frac{0.01}{t}} \left(\frac{t}{t-0.01}\right) \times \left(0.01^{1-\frac{0.01}{t}}\right),$$

$$X_3 = \Phi\left(\frac{2500-(90+78t)}{\sqrt{2.5+6t^6}}\right) - \Phi\left(\frac{2000-(90+78t)}{\sqrt{2.5+6t^6}}\right),$$

$$Y_3 = 1 - \frac{1}{100}\left[\left(\frac{22}{13}\right)^{\frac{0.01}{t}} + (0.4)^{\frac{0.01}{t}}\right] \times \left(\frac{t}{t-0.01}\right)\left(0.01^{1-\frac{0.01}{t}}\right). \qquad (45.22)$$

Figure 45.10 shows the probability that the system is in state 0 versus the time t, where the solid line represents the compound Poisson process $D(t) = \sum_{i=0}^{N(t)} X_i$ with rate $\lambda = 0.04$, and the dotted line represents the compound Poisson process with rate $\lambda = 0.8$. In Fig. 45.10, we observe that the probability of being in state 0 is close to zero when $t > 100$ for the rate $\lambda = 0.8$.

The probability of being in state F is calculated as:

$$\begin{aligned} P_t(F) &= P\left[Y_1(t) \le G_1, Y_2(t) \le G_2, D(t) > S\right] \\ &= KL\left[1 - \mathrm{e}^{-\lambda_2 t} \sum_{j=0}^{\infty} \left(\frac{\lambda_2 t}{j!}\right) F_X^{(j)}(200)\right], \end{aligned}$$

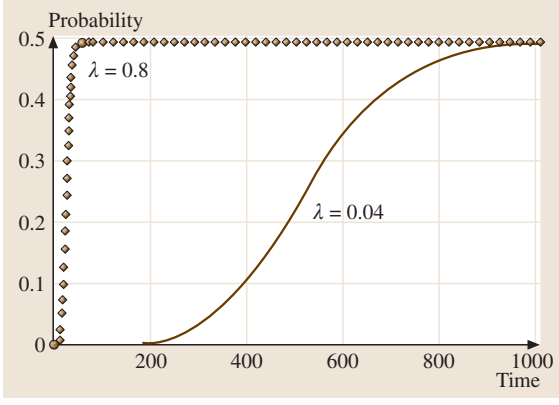


Fig. 45.11 Probability plot for state F versus time

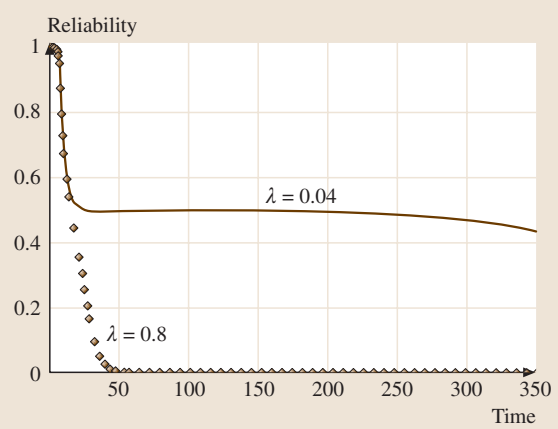


Fig. 45.12 Reliability versus time

$$\text{where}\quad K=\Phi\left(\frac{2500-(90+78t)}{\sqrt{2.5+6t^6}}\right),$$
$$L=1-\frac{1}{100}(0.4)^{\frac{0.01}{t}}\left(\frac{t}{t-0.01}\right)\left(0.01^{1-\frac{0.01}{t}}\right). \tag{45.23}$$

Figure 45.11 shows the probability of being in state F as a function of time t, where the solid line represents the compound Poisson process $D(t)=\sum_{i=0}^{N(t)}X_i$ with rate $\lambda=0.04$, and the dotted line represents the compound Poisson process with rate $\lambda=0.8$.

Finally, the system reliability $R(t)$ is given by

$$\begin{aligned}R(t)&=P(\text{system state}\geq 1)\\&=\sum_{i=1}^{3}P_t(i)\\&=X_3Y_3\,\mathrm{e}^{-\lambda_2 t}\sum_{j=0}^{\infty}\left(\frac{\lambda_2 t}{j!}\right)F_X^{(j)}(200),\end{aligned}$$

$$\text{where } X_3=\Phi\left(\frac{2000-(90+78t)}{\sqrt{2.5+6t^6}}\right)$$
$$\times\left\{1-\frac{1}{100}\left[(0.4)^{\frac{0.01}{t}}+\left(\frac{22}{13}\right)^{\frac{0.01}{t}}\right]\right.$$
$$\left.\times\left(\frac{t}{t-0.01}\right)\left(0.01^{1-\frac{0.01}{t}}\right)\right\},$$
$$Y_3=\left[\Phi\left(\frac{2500-(90+78t)}{\sqrt{2.5+6t^6}}\right)\right.$$
$$\left.-\Phi\left(\frac{2000-(90+78t)}{\sqrt{2.5+6t^6}}\right)\right]$$
$$\times\left[1-\frac{1}{100}\left(\frac{t}{t-0.01}\right)\right.$$
$$\left.\times\left(\frac{22}{13}\right)^{\frac{0.01}{t}}0.01^{1-\frac{0.01}{t}}\right]. \tag{45.24}$$

Figure 45.12 shows the system reliability versus time t, where the solid line represents the compound Poisson process with rate $\lambda=0.04$, and the dotted line represents the compound Poisson process with rate $\lambda=0.8$. As for the rate $\lambda=0.8$ we observe that the system will probably fail after a time t of 50. It seems that the random shock process governs the behavior of the reliability function. Therefore, the dotted line quickly approaches the failure caused by the shock damage.

45.3 Repairable Degraded Systems Modeling

45.3.1 Inspection–Maintenance Model Subject to Two Competing Processes

Model description

Assumptions. The system starts in a new condition. The assumptions are as follows [45.22]:

1. The system is not continuously monitored, its state can be detected only by inspection, but system failure is self-announcing without inspection.
2. After a PM or CM action, the system will be restored back to an as-good-as-new state.
3. A CM action is more costly than a PM, and a PM costs much more than an inspection. This implies $C_c > C_p > C_i$.
4. The two processes $Y(t)$ and $D(t)$ are independent.
5. Repair time is not negligible.

Although continuous monitoring processes are feasible for some systems, the cost to monitor the process and the

labor required would, however, not make it realistic in practice. Therefore, we need to improve the system performance by determining the periodic inspections with maintenance action that will minimize the average total system maintenance cost. Since system deterioration while running leads to system failure, it proves better to assume that the degradation paths are continuous and increasing functions.

Inspection–Maintenance Policy. It is proposed that the system is periodically inspected at times $\{I, 2I, \cdots, nI, \cdots\}$. We assume that the degradation ($\{Y(t)\}_{t\geq 0}$) and random shock processes ($\{D(t)\}_{t\geq 0}$) are independent. Let T denote the time to failure, defined as

$$T = \inf\left[t > 0 : Y(t) > G \text{ or } D(t) > S\right] ,$$

where G is the critical value for $\{Y(t)\}_{t\geq 0}$ and S is the threshold level for $\{D(t)\}_{t\geq 0}$.

The two threshold values L and G (where G is fixed) effectively divide the system state into three regions, as illustrated in Fig. 45.13. They are: the doing-nothing zone; the PM zone; and the CM zone. The maintenance action will be performed when either of the following situations occurs.

1. The current inspection reveals that the system condition falls into the PM zone, and this state was not found on previous inspection. At inspection time iI, the system falls into the PM zone, which means $\{Y[(i-1)I] \leq L, D[(i-1)I] \leq S\} \cap \{L < Y(iI) \leq G, D(iI) \leq S\}$. Then PM action is performed and will take a random time R_1.

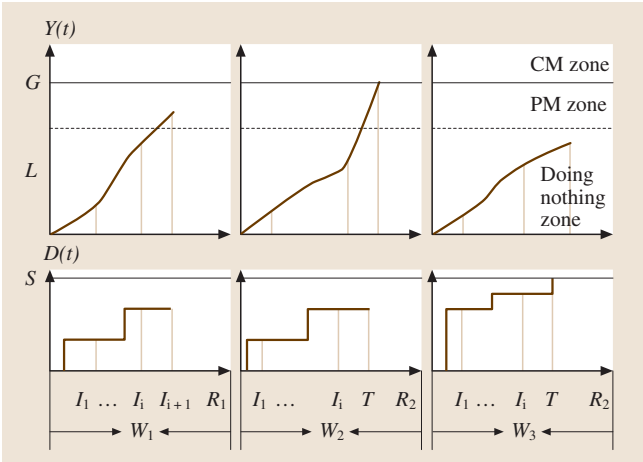


Fig. 45.13 The evolution of the system

2. When the system fails at T, a CM action is taken immediately and takes time R_2.

It is assumed that both PM and CM actions are considered to be perfect. Even though both PM and CM actions bring the system back to an as-good-as-new state, they are, physically, not necessarily the same, since a CM has to performed on a worse system. Hence, CM is likely to be more complex and expensive. Therefore, it is realistic to assume that the repair time is not negligible. This chapter considers that the PM action will take a random amount of time R_1 and that a CM action will take a random amount of time R_2. After a PM or a CM action is performed, the system is renewed. A new sequence of the inspection would start again, defined in the same way.

Maintenance Cost Modeling

In this section, an explicit expression for the average long-run maintenance cost per unit time is derived. The objectives of the model are to determine the optimal PM threshold L and the optimal inspection time I. From the basics of renewal reward theory, we have

$$\lim_{t\to\infty} \frac{C(t)}{t} = \frac{E[C_1]}{E[W_1]} .$$

We now model the average total maintenance cost per unit time on a single renewal cycle instead of $\lim_{t\to\infty} \frac{C(t)}{t}$; then we will analyze $E[C_1]$ and $E[W_1]$.

Expected maintenance cost analysis in a cycle. The expected total maintenance cost during a cycle $E[C_1]$ is expressed as [45.22]:

$$E[C_1] = C_i E[N_I] + C_p E[R_1] P_p + C_c E[R_2] P_c . \quad (45.25)$$

During a renewal cycle, activities in terms of costs include: inspection cost, time to repair, and PM or CM actions. The renewal cycle will end by either a PM or a CM action. With a probability of P_p, the cycle will end with a PM action and it will take on average an amount of time $E[R_1]$ to complete a PM action, with a corresponding cost of $C_p E[R_1] P_p$. Similarly, if a cycle ends with a CM action with probability P_c, it will take on average an amount of time $E[R_2]$ to complete a CM action, with a corresponding cost of $C_c E[R_2] P_c$. In the following, we will perform the analysis of $E[C_1]$.

Calculate $E[N_I]$. Let $E[N_I]$ denote the expected number of inspections during a cycle. $E[N_I]$ can be obtained

as:

$$E[N_I] = \sum_{i=1}^{\infty} (i) P(N_I = i) \,. \tag{45.26}$$

Obviously $\sum_{i=1}^{\infty} P(N_I = i) = 1$. There will be a total of i inspections during a cycle if the first PM trigger falls within the time interval $[(i-1)I, iI]$, or if the system condition is in the doing-nothing zone before time iI and the system fails during the interval $[iI, (i+1)I]$. In other words, the inspection will stop when the i-th inspection finds that a PM condition is satisfied while this situation was not revealed in the previous inspection, or the system fails during the interval $[iI < T \le (i+1)I]$ while the system is in the doing-nothing zone before iI.

Let $P(N_I = i)$ denote the probability that there a total of i inspections occur in a renewal cycle. Then we have

$$\begin{aligned} P(N_I = i) = {} & P\{Y[(i-1)I] \le L, D[(i-1)I] \le S\} \\ & \times P[L < Y(iI) \le G, D(iI) \le S] \\ & + P[Y(iI) \le L, D(iI) \le S] \\ & \times P[iI < T \le (i+1)I] \,. \end{aligned} \tag{45.27}$$

Hence,

$$\begin{aligned} E[N_I] = {} & \sum_{i=1}^{\infty} i\{P\{Y[(i-1)I] \le L, D[(i-1)I] \le S\} \\ & \times P[L < Y(iI) \le G, D(iI) \le S] \\ & + P[Y(iI) \le L, D(iI) \le S] \\ & \times P[iI < T \le (i+1)I]\} \,. \end{aligned} \tag{45.28}$$

We now calculate the probabilities $P\{Y[(i-1)I] \le L, D[(i-1)I] \le S\}$ and $P[L < Y(iI) \le G, D(iI) \le S]$ with the following two different expressions for $Y(t)$.

A) Assume $Y(t) = A + Bg(t)$ where $A \sim N\left(\mu_A, \sigma_A^2\right)$, $B \sim N\left(\mu_B, \sigma_B^2\right)$, and A and B are independent. Given $g(t) = t$. $D(t) = \sum_{i=0}^{N(t)} X_i$ where the X_i are iid and $N(t) \sim Poisson(\lambda)$. Then

$$\begin{aligned} & P\{Y[(i-1)I] \le L, D[(i-1)I] \le S\} \\ & = P[A + B(i-1)I \le L] \\ & \quad \times P\left\{D[(i-1)I] = \sum_{i=0}^{N[(i-1)I]} X_i \le S\right\} \\ & = \Phi\left(\frac{L - (\mu_A + \mu_B(i-1)I)}{\sqrt{\sigma_A^2 + \sigma_B^2((i-1)I)^2}}\right) \mathrm{e}^{-\lambda(i-1)I} \\ & \quad \times \sum_{j=0}^{\infty} \frac{(\lambda(i-1)I)^j}{j!} F_X^{(j)}(S) \end{aligned} \tag{45.29}$$

and

$$\begin{aligned} & P[L < Y(iI) \le G, D(iI) \le S] \\ & = \left[\Phi\left(\frac{G - (\mu_A + \mu_B iI)}{\sqrt{\sigma_A^2 + \sigma_B^2(iI)^2}}\right) \right. \\ & \quad \left. - \Phi\left(\frac{L - (\mu_A + \mu_B iI)}{\sqrt{\sigma_A^2 + \sigma_B^2(iI)^2}}\right)\right] \mathrm{e}^{-\lambda iI} \\ & \quad \times \sum_{j=0}^{\infty} \frac{(\lambda iI)^j}{j!} F_X^{(j)}(S) \,. \end{aligned} \tag{45.30}$$

B) Assume $Y(t) = W\frac{\mathrm{e}^{Bt}}{A + \mathrm{e}^{Bt}}$, where W is a constant, $A \sim U[0, a]$, $a > 0$; $B \sim \mathrm{Exp}(\beta)$, $\beta > 0$, A and B are independent. $D(t) = \sum_{i=0}^{N(t)} X_i$ where the X_i are iid and $N(t) \sim Possion(\lambda)$. Then

$$\begin{aligned} & P\{Y[(i-1)I] \le L, D[(i-1)I] \le S\} \\ & = \left[1 - \frac{1}{a}\left(\frac{1-u_1}{u_1}\right)^{\frac{\beta}{I_{i-1}}} \left(\frac{(i-1)I}{(i-1)I - \beta}\right) \right. \\ & \quad \left. \times \left(a^{1 - \frac{\beta}{(i-1)I_1}} - 1\right)\right] \mathrm{e}^{-\lambda(i-1)I} \\ & \quad \times \sum_{j=0}^{\infty} \frac{[\lambda(i-1)I]^j}{j!} F_X^{(j)}(S) \,, \end{aligned} \tag{45.31}$$

where $u_1 = L/W$. Similarly,

$$\begin{aligned} & P[L < Y(iI) \le G, D(iI) \le S] \\ & = \left\{\frac{1}{a}\left(\frac{iI}{iI - \beta}\right) \right. \\ & \quad \left. \times \left(a^{1 - \frac{\beta}{iI}}\right)\left[\left(\frac{1-u_3}{u_3}\right)^{\frac{\beta}{iI}} - \left(\frac{1-u_2}{u_2}\right)^{\frac{\beta}{iI}}\right]\right\} \mathrm{e}^{-\lambda iI} \\ & \quad \times \sum_{j=0}^{\infty} \frac{(\lambda iI)^j}{j!} F_X^{(j)}(S) \,, \end{aligned} \tag{45.32}$$

where $u_2 = G/W$, $u_3 = L/W$.

Secondly, we discuss the calculation of $P[iI < T \le (i+1)I]$. The definition of T is $T = \inf[t > 0 : Y(t) > G \text{ or } D(t) > S]$. According to the definition, we derive the expression:

$$\begin{aligned} & P[iI < T \le (i+1)I] \\ & = P\{Y(iI) \le L, Y[(i+1)I] > G\} \\ & \quad \times P\{D[(i+1)I] \le S\} \\ & \quad + P\{Y[(i+1)I] \le L\} \\ & \quad \times P\{D(iI) \le S, D[(i+1)I] > S\} \,. \end{aligned} \tag{45.33}$$

In (45.33), since $Y(iI)$, $Y[(i+1)I]$ are not independent, we could obtain the joint pdf $f_{Y(iI),Y[(i+1)I]}(y_1, y_2)$ in order to compute $P\{Y(iI) \le L, Y[(i+1)I] > G\}$. We consider two different expressions for $Y(t)$. The details are as follows:

A) Assume $Y(t) = A + Bg(t)$ where $A > 0$ and $B > 0$ are two independent random variables, and $g(t)$ is an increasing function of time t. Assume that $A \sim f_A(a)$, $B \sim f_B(b)$. Let

$$\begin{cases} y_1 = a + bg(iI) \\ y_2 = a + bg[(i+1)I] \end{cases} .$$

After simultaneously solving the above equations in terms of y_1 and y_2, we obtain:

$$a = \frac{y_1 g[(i+1)I] - y_2 g(iI)}{g[(i+1)I] - g(iI)} = h_1(y_1, y_2) ,$$
$$b = \frac{y_2 - y_1}{g[(i+1)I] - g(iI)} = h_2(y_1, y_2) .$$

The *Jacobian* J is given by

$$J = \begin{vmatrix} \frac{\partial h_1}{\partial y_1} & \frac{\partial h_1}{\partial y_2} \\ \frac{\partial h_2}{\partial y_1} & \frac{\partial h_2}{\partial y_2} \end{vmatrix} = \left| \frac{1}{g(iI) - g[(i+1)I]} \right| .$$

Then the random vector $\{Y(iI), Y[(i+1)I]\}$ has a joint continuous pdf as follows

$$f_{Y(iI),Y[(i+1)I]}(y_1, y_2) = |J| \, f_A[h_1(y_1, y_2)] f_B[h_2(y_1, y_2)] . \quad (45.34)$$

B) Assume $Y(t) = \frac{W e^{At}}{B + e^{At}}$ where $A > 0$ and $B > 0$ are independent. Assume $A \sim f_A(a)$, $B \sim f_B(b)$. Let

$$\begin{cases} y_1 = \frac{W e^{aiI}}{b + e^{aiI}} \\ y_2 = \frac{W e^{a(i+1)I}}{b + e^{a(i+1)I}} \end{cases} .$$

The solutions for a and b can be easily found from the above equations in terms of y_1 and y_2 as follows:

$$\begin{cases} a = \frac{\ln\left(\frac{y_2(y_1 - W)}{y_1(y_2 - W)}\right)}{I} = h_1(y_1, y_2) \\ b = -\frac{e^{\frac{\ln\left(\frac{y_2(y_1 - W)}{y_1(y_2 - W)}\right)(i+1)I}{I}}}{y_2} (y_2 - W) = h_2(y_1, y_2) \end{cases} .$$

It can be shown that the random vector $\{Y(iI), Y[(i+1)I]\}$ has a joint density function given by

$$f_{Y(iI),Y[(i+1)I]}(y_1, y_2) = |J| \, f_A[h_1(y_1, y_2)] f_B[h_2(y_1, y_2)] , \quad (45.35)$$

where the Jacobian determinant J is given in Appendix A.

As for the term $P\{D(iI) \le S, D[(i+1)I] > S\}$ in (45.30), since $D(t) = \sum_{i=0}^{N(t)} X_i$ is a compound Poisson process, the compound Poisson process has a stationary independent increment property. Therefore, the random variables $D(iI)$ and $D[(i+1)I] - D(iI)$ are independent. Using the *Jacobian* transformation, the random vector $\{D(iI), D[(i+1)I] - D(iI)\}$ is distributed in the same way as vector $\{D(iI), D[(i+1)I]\}$. Note that $D(iI)$ and $D(I_{i+1})$ are independent, therefore,

$$P\{D(iI) \le S, D[(i+1)I] > S\} = P[D(iI) \le S] P\{D[(i+1)I] > S\} . \quad (45.36)$$

Calculate P_p. Note that either a PM or CM action will end a renewal cycle. In other words, these two events are mutually exclusive at the renewal time point. As a consequence, $P_p + P_c = 1$. The probability P_p can be obtained as follows:

$$\begin{aligned} P_p &= P(\text{PM ending a cycle}) \\ &= \sum_{i=1}^{\infty} P\{Y[(i-1)I] \le L, L < Y(iI) \le G\} \\ &\quad \times P[D(iI) \le S] . \end{aligned} \quad (45.37)$$

Analysis of expected cycle length. Since the renewal cycle ends either by a PM action with probability P_p or a CM action with probability P_c, the mean cycle length $E[W_1]$ is calculated as follows:

$$\begin{aligned} E[W_1] &= \sum_{i=1}^{\infty} E[(iI + R_1) I_{\text{PM occurs in}[(i-1)I, iI]}] \\ &\quad + E\left[(T + R_2) 1_{\text{CM occurs}}\right] \\ &= \left\{ \sum_{i=1}^{\infty} iI P\{Y[(i-1)I] \le L, \right. \\ &\quad D[(i-1)I] \le S\} P[L < Y(iI) \le G, \\ &\quad \left. D(iI) \le S] \right\} \\ &\quad + E[R_1] P_p + (E[T] + E[R_2]) P_c , \end{aligned} \quad (45.38)$$

where $I_{\text{PM occurs in}[(i-1)I, iI]}$ and $I_{\text{CM occurs}}$ are the indicator functions.

The mean time to failure, $E[T]$ is given by [45.22]:

$$\begin{aligned} E[T] &= \int_0^\infty P\{T > t\}\,\mathrm{d}t \\ &= \int_0^\infty P[Y(t) \le G,\, D(t) \le S]\,\mathrm{d}t \\ &= \int_0^\infty P[Y(t) \le G] \sum_{j=0}^{\infty} \frac{(\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t}}{j!} F_X^{(j)}(S)\,\mathrm{d}t \end{aligned}$$

or, equivalently:

$$E[T] = \sum_{j=0}^{\infty} \frac{F_X^{(j)}(S)}{j!} \int_0^\infty P[Y(t) \le G](\lambda_2 t)^j\, \mathrm{e}^{-\lambda_2 t}\,\mathrm{d}t \tag{45.39}$$

The expression $E[T]$ depends on the probability $P[Y(t) \le G]$ and cannot always be easily be obtained in closed form.

Optimization of the maintenance cost rate policy

We determine the optimal inspection time I and PM threshold L such that the long-run average maintenance cost rate $EC(L, I)$ is minimized. Mathematically, we wish to minimize the following objective function [45.22]:

$$\begin{aligned} & EC(L, I) \\ &= \frac{\sum_{i=1}^{\infty} i P_1 P_2}{\left\{\sum_{i=1}^{\infty} I_i P_1 P_2\right\} + E[R_1]P_\mathrm{p} + E[R_2]P_\mathrm{c}} \\ &\quad + \frac{\sum_{i=1}^{\infty} i V_i \{P_3 P_4 + P_5 P_6\}}{\left\{\sum_{i=1}^{\infty} I_i P_1 P_2\right\} + E[R_1]P_\mathrm{p} + E[R_2]P_\mathrm{c}} \\ &\quad + \frac{C_\mathrm{p} E[R_1] \sum_{i=1}^{\infty} P_1 P_2}{\left\{\sum_{i=1}^{\infty} I_i P_1 P_2\right\} + E[R_1]P_\mathrm{p} + E[R_2]P_\mathrm{c}} \\ &\quad + \frac{C_\mathrm{c} E[R_2]\left\{1 - \sum_{i=1}^{\infty} P_1 P_2\right\}}{\left\{\sum_{i=1}^{\infty} I_i P_1 P_2\right\} + E[R_1]P_\mathrm{p} + E[R_2]P_\mathrm{c}}\,, \end{aligned} \tag{45.40}$$

where $I_{i-1} = (i-1)I$, $I_i = iI$, $I_{i+1} = (i+1)I$ and $V_i = P[Y(iI) \le L, D(iI) \le S]$, P_1: $P[Y(I_{i-1}) \le L, D(I_{i-1}) \le S]$, P_2: $P[L < Y(I_i) \le G, D(I_i) \le S]$, P_3: $P[Y(I_i) \le L, Y(I_{i+1}) > G]$, P_4: $P[D(I_{i+1}) \le S]$, P_5: $P[Y(I_{i+1}) \le L]$, P_6: $P[D(I_i) \le S, D(I_{i+1}) > S]$

This complex objective function is a nonlinear optimization problem and it is hard to obtain closed-form optimal solutions for L and I. *Nelder* and *Mead* [45.23] introduced a downhill simplex method that does not require the calculation of derivatives. A simplex is the most elementary geometrical scheme that can be formed in n dimensions and has $(n+1)$ vertices. A brief summary of the steps of the method is: each iteration generates a new vertex for the simplex. If the new point is better than at least one of the existing vertices, it then replaces the worst vertex. The search direction is generated through reflection, expansion and contraction operations.

A step-by-step algorithm proposed by *Li* and *Pham* [45.21] based on the Nelder–Mead downhill simplex method is summarized as follows:

- Step 1: choose (n+1) distinct vertices as an initial set $\left\{Z^{(1)}, \cdots, Z^{(n+1)}\right\}$. Then calculate the function value $f(Z)$ for $i = 1, 2, \ldots, (n+1)$, where $f(Z) = EC(I, L)$. Put the values $f(Z)$ in an increasing order where $f(Z^{(1)}) = \min[EC(I, L)]$ and $f\left(Z^{(n+1)}\right) = \max[EC(I, L)]$. Set $k = 0$.
- Step 2: compute the best-n centroid $X^{(k)} = \frac{1}{n}\sum_{i=1}^{n} Z^{(i)}$.
- Step 3: use the centroid $X^{(k)}$ in Step 2 to compute the away-from-worst move direction

$$\Delta X^{(k+1)} = X^{(k)} - Z^{(n+1)}\,.$$

- Step 4: set $\lambda = 1$ and compute $f(X^{(k)} + \lambda\Delta X^{(k+1)})$. If $f(X^{(k)} + \lambda\Delta X^{(k+1)}) \le f(Z^{(1)})$ then go to Step 5. Otherwise, if $f(X^{(k)} + \lambda\Delta X^{(k+1)}) \ge f(Z^{(n)})$ then go to Step 6. Otherwise, fix $\lambda = 1$ and go to Step 8.
- Step 5: Set $\lambda = 2$ and compute $f(X^{(k)} + 2\Delta X^{(k+1)})$. If $f(X^{(k)} + 2\Delta X^{(k+1)}) \le f(X^{(k)} + \Delta X^{(k+1)})$ then set $\lambda = 2$. Otherwise, set $\lambda = 1$. Then go to Step 8.
- Step 6: If $f(X^{(k)} + \lambda\Delta X^{(k+1)}) \le f(Z^{(n+1)})$ then set $\lambda = 1/2$. Compute $f(X^{(k)} + \frac{1}{2}\Delta X^{(k+1)})$. If $f(X^{(k)} + \frac{1}{2}\Delta X^{(k+1)}) \le f(Z^{(n+1)})$ then set $\lambda = 1/2$ and go to Step 8. Otherwise, set $\lambda = -1/2$ and, if $f(X^{(k)} - \frac{1}{2}\Delta X^{(k+1)}) \le f(Z^{(n+1)})$, then set $\lambda = -1/2$ and go to Step 8. Otherwise, go to Step 7.
- Step 7: shrink the current solution set toward the best $Z^{(1)}$ by $Z^{(i)} = \frac{1}{2}(Z^{(1)} + Z^{(i)})$, $i = 2, \cdots, n+1$. Compute the new $f(Z^{(2)}), \cdots, f(Z^{(n+1)})$, let $k = k+1$, and return to Step 2.
- Step 8: Replace the worst $Z^{(n+1)}$ by $X^{(k)} + \lambda\Delta X^{(k+1)}$. If $\sqrt{\frac{1}{n+1}\sum_{i=1}^{n+1}[f(Z^{(i)}) - \overline{f}\,]^2} < 0.5$, where $\overline{f}$ is an average value, then STOP. Otherwise, let $k = k+1$ and return to Step 2. (It should be noted that the criterion in Step 8 is not unique but will depend on how soon you would like the algorithm to stop when the function values at the vertices are close. Here we do this when the difference be-

tween the maximum and the minimum values of f is less than 0.5.)

A Numerical Example

Here we present an example to illustrate the results and the step-by-step application procedure.

Assume that the degradation process is described by $Y(t) = A + Bg(t)$, where A and B are independent and follow a uniform distribution with parameter interval [0,4] and an exponential distribution with parameter 0.3, i. e., $A \sim U(0, 4)$ and $B \sim \text{Exp}(-0.3t)$, respectively, and $g(t) = \sqrt{t}\,\text{e}^{0.005t}$.

Assume that the random shock damage is described by $D(t) = \sum_{i=1}^{N(t)} X_i$, where X_i follows an exponential distribution, i. e., $X_i \sim \text{Exp}(-0.04t)$ and $N(t) \sim Poisson(0.1)$. Also $G = 50$, $S = 100$, $C_i = 900$/inspection, $C_c = 5600$/CM, $C_p = 3000$/PM, $R_1 \sim \text{Exp}(-0.1t)$, and $R_2 \sim \text{Exp}(-0.04t)$. We now determine both the values of I and L so that the average total cost per unit time $EC(I, L)$ is minimized. Following are step-by-step procedure [45.22]:

- Step 1: since there are two decision variables I and L, we need $(n+1) = 3$ initial distinct vertices, which are $Z^{(1)} = (25, 20)$, $Z^{(2)} = (20, 18)$, and $Z^{(3)} = (15, 10)$. Set $k = 0$. We calculate the value of $f\left(Z^{(\cdot)}\right)$ corresponding to each vertex and sort them in increasing order of $EC(I, L)$.
- Step 2: calculate the centroid: $X^{(0)} = \left(Z^{(1)} + Z^{(1)}\right)/2 = (22.5, 19)$.
- Step 3: generate the search direction: $\Delta X = X^{(0)} - Z^{(2)} = (7.5, 9)$.
- Step 4: set $\lambda = 1$, which will produce a new minimal $EC(30, 28) = 501.76$ that leads us to try an expansion with $\lambda = 2$, that is (37.5, 38).
- Step 5: set $\lambda = 2$. Similarly, calculate $f(Z)$, which leads to $EC(37.5, 38) = 440.7$. Go to Step 8. This result turns out to be a better solution, hence (15, 10) is replaced by (37.5, 38).

The iteration continues and stops at $k = 6$ (Table 45.1) since $\sqrt{\frac{1}{3}\sum_{i=1}^{3}\left[EC(Z^{(i)}) - \overline{EC(I, L)}\right]^2} < 0.5$, where $\overline{EC(I, L)}$ is the average value.

Table 45.1 illustrates the process of the Nelder–Mead algorithm. In Table 45.1, $Z^{(\cdot)} = (I, L)$. From Table 45.1, we observe that a set of the optimal values is

$$I^* = 37.5,\ L^* = 38$$

and the corresponding cost value is $EC^*(I, L) = 440.7$.

Table 45.2 illustrates the various values of L on P_c for given $I = 37.5$. From Table 45.2, we observe that the probability P_c increases as L increases. In other words, a larger value for L will put the system at high risk of failure.

Figure 45.14 shows the relationship between L and P_c for different I values, such as $I = 35$, $I = 37.5$, and $I = 40$. From Fig. 45.14, we observe that P_c is an increasing function of L. This means a higher preventive-maintenance threshold is more likely to result in a failure.

Figure 45.15 depicts the effect of the first inspection time on P_p for various L values such as $L = 33$, $L = 35$, $L = 37$ and $L = 39$. Shorter inspection times will cause more-frequent inspection and, as

Table 45.1 Optimal values I and L

k	$Z^{(1)}$	$Z^{(2)}$	$Z^{(3)}$	Search result
0	(25,20) $EC(I, L) = 564.3$	(20,18) $EC(I, L) = 631.1$	(15,10) $EC(I, L) = 773.6$	(37.5, 38) $EC(I, L) = 440.7$
1	(37.5,38) $EC(I, L) = 440.7$	(25,20) $EC(I, L) = 564.3$	(20,18) $EC(I, L) = 631.1$	(42.5,40) $EC(I, L) = 481.2$
2	(37.5,38) $EC(I, L) = 440.7$	(42.5,40) $EC(I, L) = 481.2$	(25,20) $EC(I, L) = 564.3$	(32.5,29) $EC(I, L) = 482.2$
3	(37.5,38) $EC(I, L) = 440.7$	(42.5,40) $EC(I, L) = 481.2$	(32.5,29) $EC(I, L) = 482.2$	(32.5,33.5) $EC(I, L) = 448.9$
4	(37.5,38) $EC(I, L) = 440.7$	(32.5,33.5) $EC(I, L) = 448.9$	(42.5,40) $EC(I, L) = 481.2$	(38.75,37.125) $EC(I, L) = 441.0$
5	(37.5,38) $EC(I, L) = 440.7$	(38.75,37.125) $EC(I, L) = 441.0$	(32.5,33.5) $EC(I, L) = 448.9$	(35.3125,35.25) $EC(I, L) = 441.1$
6	(37.5,38) $EC(I^*, L^*) = 440.7$	(38.75,37.125) $EC(I, L) = 441.0$	(35.3125,35.25) $EC(I, L) = 441.4$	Stop

Table 45.2 The effect of L on P_c for $I = 37.5$

L	P_c
33	0.465
35	0.505
37	0.654
39	0.759

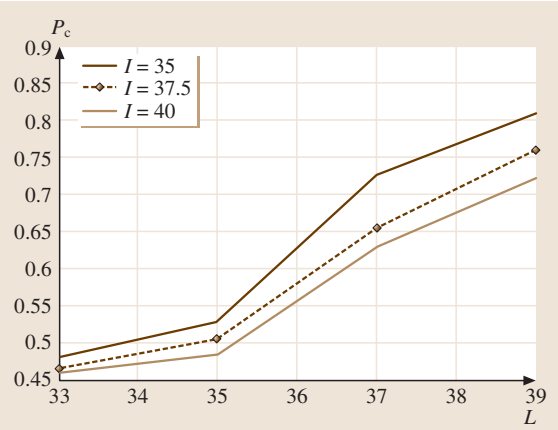


Fig. 45.14 P_c versus L

a result, will increase the probability of a PM. From Fig. 45.15, we also observe that, for smaller L values ($L = 33$ and $L = 35$), the curve decreases slightly as I increases; while, for larger values of L such as $L = 37$ and $L = 39$, the curve has a larger decrease as I increases. We also observe that the curve is more sensitive to the value of L, especially when L is large.

In summary, we observe that, on one hand, a lower value of L will result in frequent PM action and prevents full usage of the residual life of the systems. Frequent PM actions might reduce the chance of high deterioration and failures, but will also be costly. On the other hand, a higher L value will keep the system working in a higher-risk condition. Also, frequent inspections will reduce the probability of failure, while incurring additional cost.

45.3.2 Inspection–Maintenance Model for Degraded Systems with Three Competing Processes

General Inspection–Maintenance Description

This section considers systems with inspection-based maintenance subject to three failure processes that

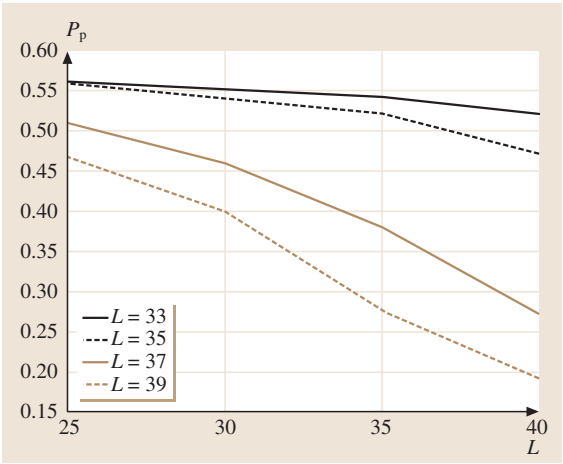


Fig. 45.15 The effect of the inspection sequence on P_p for given L

are competing for the life of such systems: two of these are degradation processes called degradation process i (measured by $Y_i(t)$ for $i = 1, 2$) and the third is a random shock process measured by the function $D(t)$ [45.21].

We assume that the three processes are independent and whichever process occurs first will cause the system to fail, where the failure of the system is defined as when $Y_1(t) > G_1$, $Y_2(t) > G_2$ or $D(t) > S$. The state of the system can only be revealed through inspection.

Assumptions.

1. System failure is only detected by inspection. Inspections are assumed to be instantaneous, perfect and nondestructive. Since the system is not continuously monitored, if the system fails it will remain failed until the next inspection, which causes a loss of C_m per unit time. In this case, a maintenance action is begun instantaneously at the inspection time.
2. After a maintenance action, either PM or CM, the system state will start as good as new.
3. A CM action will cost more than a PM action. Similarly, a PM action will cost much more than an inspection itself. This implies that $C_c > C_p > C_i$.
4. The three nondecreasing processes $Y_1(t)$, $Y_2(t)$, and $D(t)$ are independent.
5. No continuous monitoring is performed on the system.
6. The time for a CM or PM action is negligible.

We consider a system subject to three competing processes; two of them are continuous, gradual degradation processes with different characteristics, and the third is a random shock process. Applications of such systems can be found in the Space Shuttle computer complex due to critical mission phases such as boost, reentry and landing and in electric generator power systems due to the loss of commercial power systems. More related applications can be found in [45.13].

Although a continuous monitoring process is feasible for some systems, the cost of monitoring the process and the labor required would not make it realistic in practice. Therefore the criteria we consider here is to improve the system performance by performing periodic inspections, with a maintenance action if necessary, to minimize the total system maintenance cost.

Inspection–maintenance policy. The length of the inspection will be reduced as the system ages. In other words, the intervals between successive inspections become shorter as the system ages. A geometric sequence is applied in this study to develop the inter-inspection sequence. The inspection time is constructed as $I_n = \sum_{j=1}^{n} \alpha^{j-1} I_1$, where $0 < \alpha \leq 1$ and I_1 is the first inspection time. We define $U_n = I_n - I_{n-1} = \alpha^{n-1} I_1$ as the inter-inspection interval and $(U_i)_{i \in N}$ as a decreasing geometric sequence. According to the state detected at the inspection $I_n, n = 1, \cdots$, one of the following actions will happen [45.21]:

1. If both degradation values are below their PM thresholds and the shock damage value is less than its threshold, in other words $[Y_1(I_n) \leq L_1, Y_2(I_n) \leq L_2] \cap [D(I_n) \leq S]$, then the system is still in a good condition. In this case, we do nothing but determine the next inspection at $I_{n+1} = I_n + U_n$, where U_n is the inter-inspection time between the n-th and the $n+1$-th inspection interval.
2. If a degradation process falls into the PM zone $[L_i < Y_i(I_n) \leq G_i, i = 1, 2]$ and the other two processes are less than their corresponding critical thresholds, then the system calls for a PM action and it is instantaneously performed accordingly.
3. If any of the process values exceed their corresponding critical thresholds $[Y_i(t) > G_i, i = 1, 2,$ or $D(t) > S]$, then the system calls for a CM action and it is instantaneously performed. In this case, the system has failed and a CM is performed on the system.

We assume that, after a maintenance action, i. e., PM or CM, the system will be restored to as good as new. A new sequence of inspection begins, defined in the same way, and the system maintenance follows the same decision rules outlined above. Figure 45.16 shows the evolution of the system, where $Y_1(t)$ and $Y_2(t)$ represent the degradation processes 1 and 2, respectively, and $D(t)$ represents a cumulative shock damage. $(W_i)_{i \in N}$ is a renewal sequence. Figure 45.17 shows the maintenance zone projected onto the $Y_1(t)$, $Y_2(t)$ planes; G_i and L_i are the CM and PM critical thresholds for $Y_1(t)$, and $Y_2(t)$ respectively.

Maintenance cost analysis

The expected total maintenance cost per cycle, $E[C_1]$, is given as:

$$E[C_1] = C_i E[N_I] + C_p P_p + C_c P_c + C_m E[\zeta] , \tag{45.41}$$

where C_i is the cost associated with each inspection, C_p is the cost associated with a PM action, and C_c is the CM action cost. Since failure is not self-announcing and it can occur at any given instant time T within the inspection time interval $[I_i, I_{i+1}]$, the system will remain idle during the interval $[T, I_{i+1}]$. The cost coefficient C_m

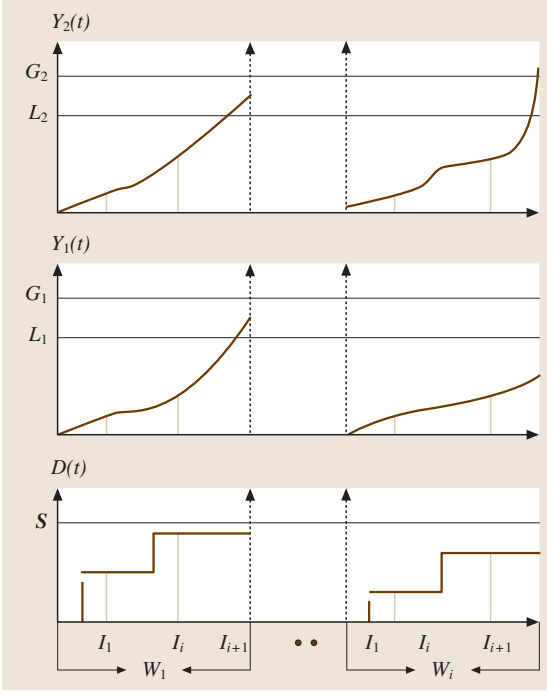


Fig. 45.16 The evolution of the system condition

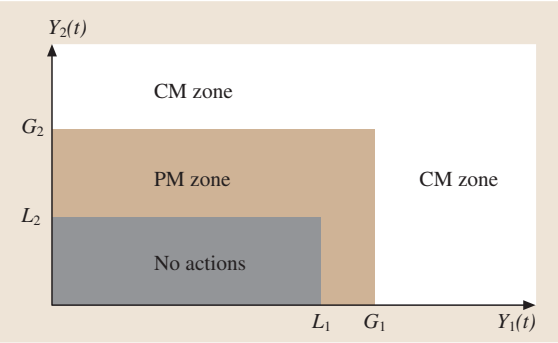


Fig. 45.17 Maintenance zone projected onto $Y_1(t)$, $Y_2(t)$

is defined as the penalty cost per unit time associated with such an event.

Calculation.

1. Let $P(N_I = i+1)$ be the probability that there are a total of $(i+1)$ inspections in the cycle. The expected number of inspections during a cycle, $E[N_I]$, is

$$E[N_I] = \sum_{i=0}^{\infty} (i+1) P_{i+1} \tag{45.42}$$

where $P_{i+1} = P(N_I = i+1)$. Note that

$$P_{i+1} = P(N_I = i+1) = \bigcup_{j=1}^{17} P\left(E_j^{(i+1)}\right) ,$$

where $E_j^{(i+1)} (j = 1, \cdots, 17)$ denotes the renewal cycle that ends at the j-th possibile time I_{i+1}. The details of all $E_j^{(i+1)}$, where the $E_j^{(i+1)}$ are mutually disjoined events for $j = 1, \cdots, 18$ are listed in Appendix B.

There are a total of 18 system state combinations revealed at any given interval $(I_i, I_{i+1}]$ where there is only one state event, $E_{18}^{(i+1)}$ (Appendix B) representing that the system is in a good condition and that no maintenance action will be required. Any other remaining state events will trigger either a PM or a CM action at time I_{i+1}.

After some simplifications, we have

$$\begin{aligned} P_{i+1} = {} & P[Y_1(I_i) \le L_1, Y_2(I_i) \le L_2, D(I_i) \le S] \\ & - P[Y_1(I_{i+1}) \le L_1, Y_2(I_{i+1}) \\ & \le L_2, D(I_{i+1}) \le S] . \end{aligned} \tag{45.43}$$

Therefore,

$$\begin{aligned} E[N_1] = \sum_{i=0}^{\infty} (i+1) \{ & P[Y_1(I_i) \le L_1, Y_2(I_i) \\ & \le L_2, D(I_i) \le S] \\ & - P[Y_1(I_{i+1}) \le L_1, Y_2(I_{i+1}) \\ & \le L_2, D(I_{i+1}) \le S] \} . \end{aligned}$$

2. There will be either a PM or CM action to end a renewal cycle. It is obvious that the two events (PM and CM) are mutually exclusive at the renewal time point: $P_p + P_c = 1$. We now calculate P_p as follows:

$$\begin{aligned} P_p &= P(\text{the cycle ends due to a PM action}) \\ &= \sum_{i=0}^{\infty} \sum_{j=1}^{3} P\left[E_j^{(i+1)}\right] . \end{aligned}$$

After some simplifications, we obtain

$$\begin{aligned} P_p = \sum_{i=0}^{\infty} \{ & P[Y_1(I_i) \le L_1, L_1 < Y_1(I_{i+1}) \le G_1] \\ & \times P[Y_2(I_i) \le L_2, Y_2(I_{i+1}) \le G_2] \\ & \times P[D(I_{i+1})] + P[Y_1(I_{i+1}) \le L_1] \\ & \times P[Y_2(I_i) \le L_2, L < Y_2(I_{i+1}) \le G_2] \\ & \times P[D(I_{i+1})] \} \end{aligned} \tag{45.44}$$

and $P_c = 1 - P_p$. We can obtain the joint probability density function $f_{Y(I_i),Y(I_{i+1})}(y_1, y_2)$ of $Y(I_i)$ and $Y(I_{i+1})$ by computing $P[Y_1(I_i) \le L_1, Y_1(I_{i+1}) \le G_1]$ and $P[Y_2(I_i) \le L_2, Y_2(I_{i+1}) \le G_2]$.

3. Let T denote the time to failure. That is $T = \inf[t : Y_1(t) > G_1, Y_2(t) > G_2 \text{ or } D(t) > S]$. If $I_i < T \le I_{i+1}$, the unit will be idle during the interval $[T, I_{i+1}]$. Let $E[\zeta]$ denote the average idle time between the failure occurrence epoch and its inspection during the cycle. Then $E[\xi]$ is calculated as follows:

$$\begin{aligned} E[\xi] &= \sum_{i=0}^{\infty} E\left[(I_{i+1} - T) 1_{I_i < T \le I_{i+1}}\right] \\ &= \sum_{j=0}^{\infty} R_j \int_{I_i}^{I_{i+1}} (I_{i+1} - t) \mathrm{d}F_T(t) \end{aligned} \tag{45.45}$$

where

$$
\begin{aligned}
R_j = \big\{ & P[Y_1(I_i) \le L_1, L_1 < Y_1(I_{i+1}) \le G_1] \\
& \times P[Y_2(I_i) \le L_2, L_1 < Y_1(I_{i+1})] \\
& + P[Y_2(I_i) \le L_2] \\
& \times P[Y_1(I_i) \le L_1, Y_1(I_{i+1}) > G_1] \\
& + P[Y_1(I_{i+1}) \le L_1] \\
& \times P[Y_2(I_i) \le L_2] \big\} P[D(I_i) \le S] \\
F(t) = {} & P[Y_1(t) > G_1, Y_2(t) \le G_2, D(I_i) \le S] \\
& + P[Y_1(t) \le G_1, Y_2(t) > G_2, D(I_i) \le S] \\
& + P[Y_1(t) \le G_1, Y_2(t) \le G_2, D(I_i) > S]
\end{aligned}
$$

and $1_{I_i < T \le I_{i+1}}$ is an indicator function.

Expected cycle length. The expected cycle length $E[W_1]$ is given as follows:

$$
\begin{aligned}
E[W_1] &= E[E[W_1|N_I]] \\
&= \sum_{i=0}^{\infty} E[W_1|N_I = i] P(N_i = i) \\
&= \sum_{i=0}^{\infty} I_{i+1} P_{i+1} \,, \qquad (45.46)
\end{aligned}
$$

where $P_{i+1}$is given in (45.43).

Therefore, the average long-run maintenance cost rate function $EC(L_1, L_2, I_1)$ is a function of the inspection times $\{I_1, \cdots, I_i, \cdots\}$ and the PM critical threshold values (L_1, L_2) through the functions P_p, P_c, $E[N_I]$, $E[\zeta]$ and $E[W_1]$. The average long-run maintenance cost rate is, in other words, $EC(L_1, L_2, I_1) = \frac{E[C_1]}{E[W_1]}$, and can be obtained by computing the two functions given in (45.41) and (45.46).

Optimization of the maintenance cost rate

The geometric inspection sequence $\{I_1, \cdots, I_i, \cdots\}$, where $I_n = \sum_{j=1}^{n} \alpha^{j-1} I_1$, depends on I_1 for given α. In this section, we develop a step-by-step algorithm based on the Nelder–Mead downhill simplex method to obtain the optimum decision variables (I_1, L_1, L_2) such that the long-run average maintenance cost rate $EC(L_1, L_2, I_1)$ is minimized. Mathematically, the optimization problem of the cost rate function can be formulated as follows [45.21]:

Optimization problem. Find I_1, L_1 and $L_2 (0 < L_1 \le G_1, 0 < L_2 \le G_2)$ such that

$$
\begin{aligned}
EC(L_1, L_2, I_1) = {} & \frac{C_\mathrm{i} \sum_{i=0}^{\infty} (i+1) \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le S \right] - P\left[Y_1\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\}}{\sum_{i=0}^{\infty} \left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le S \right] - P\left[Y_1\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\}} \\
& + \frac{C_\mathrm{p} \sum_{i=0}^{\infty} \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le G_2 \right] P\left[Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_2, Y_2\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le G_2 \right] P\left[D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\}}{\sum_{i=0}^{\infty} \left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le S \right] - P\left[Y_1\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\}} \\
& + \frac{C_\mathrm{c} \left(1 - \sum_{i=0}^{\infty} \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le G_2 \right] P\left[Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le G_2 \right] P\left[D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\} \right)}{\sum_{i=0}^{\infty} \left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le S \right] - P\left[Y_1\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\}} \\
& + \frac{C_\mathrm{m} \sum_{i=0}^{\infty} \left\{ (R_{1i} + R_{2i} + R_{3i}) P\left[D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\} \int_{\sum_{j=1}^{i} \alpha^{j-1} I_1}^{\sum_{j=1}^{i+1} \alpha^{j-1} I_1} \left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1 - t\right) \mathrm{d}F_\mathrm{T}(t)}{\sum_{i=0}^{\infty} \left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \left\{ P\left[Y_1\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i} \alpha^{j-1} I_1\right) \le S \right] - P\left[Y_1\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_1, Y_2\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le L_2, D\left(\sum_{j=1}^{i+1} \alpha^{j-1} I_1\right) \le S \right] \right\}}
\end{aligned}
$$

is minimum, where

$$R_{1i} = P\left[Y_1\left(\sum_{j=1}^{i}\alpha^{j-1}I_1\right) \le L_1,\ L_1 < Y_1\left(\sum_{j=1}^{i+1}\alpha^{j-1}I_1\right) \le G_1\right] \times P\left[Y_2\left(\sum_{j=1}^{i}\alpha^{j-1}I_1\right) \le L_2,\ L_1 < Y_1\left(\sum_{j=1}^{i+1}\alpha^{j-1}I_1\right)\right],$$

$$R_{2i} = P\left[Y_1\left(\sum_{j=1}^{i}\alpha^{j-1}I_1\right) \le L_1,\ G_1 < Y_1\left(\sum_{j=1}^{i+1}\alpha^{j-1}I_1\right)\right] \times P\left[Y_2\left(\sum_{j=1}^{i}\alpha^{j-1}I_1\right) \le L_2\right],$$

$$R_{3i} = P\left[Y_1\left(\sum_{j=1}^{i}\alpha^{j-1}I_1\right) \le L_1\right] \times P\left[Y_2\left(\sum_{j=1}^{i}\alpha^{j-1}I_1\right) \le L_2\right].$$

This optimization function is a complex nonlinear function, the optimum solution of which is difficult to find. The Nelder–Mead downhill simplex method (discussed in Sect. 45.3.1) is the most popular direct-search method for obtaining the optimum solution of an unconstrained nonlinear function, and does not require the calculation of derivatives.

Numerical examples

This section illustrates the results in the Sect. 45.3.2. Assume that degradation process 1 is described as the function $Y_1(t) = \frac{W e^{B_1 t}}{A_1 + e^{B_1 t}}$, where the random variables A_1 and B_1 are independent and follow a uniform distribution with parameter interval [0,40], and exponential distribution with parameter 1, respectively. In short, $A_1 \sim U[0, 40]$ and $B_1 \sim \text{Exp}(1)$.

Similarly, assume that degradation process 2 is modeled as $Y_2(t) = A_2 + B_2 g(t)$ where $A_2 \sim U[0, 2]$, $B_2 \sim \text{Exp}(0.2)$ and $g(t) = \sqrt{t}e^{0.01t}$. Assume that the random shock is represented by the function $D(t) = \sum_{i=0}^{N_2(t)} X_i$, where $X_i \sim \text{Exp}(0.04)$ and $N(t) \sim Poisson(0.1)$. Also $G_1 = 300$, $G_2 = 70$ and $S = 100$.

Assume that the cost parameters are as follows: $C_c = 560$ units/CM, $C_p = 400$ units/PM, $C_i = 100$ units/inspection, $C_m = 500$ units/unit time and $\alpha = 0.97$.

The inspection sequence $\{I_1, \cdots, I_n, \cdots\}$ is constructed with $I_n = \sum_{j=1}^{n} \alpha^{j-1} I_1$. We want to determine the values of I_1 and (L_1, L_2) so that the average long-run maintenance cost rate per unit time is minimized.

Following are step-by-steps using our proposed algorithm in Sect. 45.3.1:

- Step 1: there are three decision variables, say L_1, L_2, and I_1, so we need four distinct vertices as an initial set of values: $Z^{(1)} = (270, 56, 76)$, $Z^{(1)} = (280, 60, 72)$, $Z^{(2)} = (290, 52, 66)$ and $Z^{(3)} = (300, 50, 57)$. Set $k = 0$.

We now calculate the function value $f(Z)$ corresponding to each vertex and put them in increasing order of the objective value $EC(L_1, L_2, I_1)$ from smallest to highest.

- Step 2: compute the centroid: $X^{(0)} = \frac{1}{3}\left(Z^{(1)} + Z^{(2)} + Z^{(3)}\right) = (280, 56, 71.3)$.
- Step 3: search for the away-from-worst direction: $\Delta X = X^{(0)} - Z^{(4)} = (-20, 6, 14.3)$.
- Step 4: set $\lambda = 1$, which will generate a new minimal $EC(260, 60, 85.6) = 291.9$ that leads to an expansion with $\lambda = 2$ that is (240, 60, 99.9).
- Step 5: set $\lambda = 2$. Similarly, compute $f(Z)$, which leads to 247.9. Go to Step 8

This result turned out to be a better solution, hence (300, 50, 57) is replaced by (240, 60, 99.9).

The iteration continues and stops at $k = 4$ (see Table 45.3) since

$$\sqrt{\frac{1}{4}\sum_{i=1}^{4}\left[EC(Z^{(i)}) - \overline{EC(L_1, L_2, I_1)}\right]^2} = 0.449 < 0.5\,,$$

where $\overline{EC(L_1, L_2, I_1)}$ is the average value.

From Table 45.3, we obtain the optimal solution for (L_1, L_2, I_1) as: $\left(L_1^* = 172, L_2^* = 60, I_1^* = 144\right)$ and the corresponding average long-run maintenance cost rate is $EC\left(L_1^*, L_2^*, I_1^*\right) = 245.9$. Figure 45.18 depicts the average long-run maintenance cost-rate curve $EC(L_1, L_2, I_1)$ as a function of the inspection time interval I_1 for $L_1 = 172$ and $L_2 = 60$.

Table 45.4 presents a sensitivity analysis in terms of the probability that the cycle will end due to a PM action, P_p, for various values of (L_1, L_2) for $\alpha = 0.97$

Table 45.3 Nelder–Mead algorithm results

k	$Z^{(1)} = (L_1, L_2, I_1)$	$Z^{(2)}$	$Z^{(3)}$	$Z^{(4)}$	Search result
0	(270,56,76) $\frac{E[C_1]}{E[W_1]} = 300.7$	(280,60,72) $\frac{E[C_1]}{E[W_1]} = 332.2$	(290,52,66) $\frac{E[C_1]}{E[W_1]} = 360.4$	(300,50,57) $\frac{E[C_1]}{E[W_1]} = 388.2$	$\lambda = 2$ $\frac{E[C_1]}{E[W_1]} = 247.9$
1	(240,60,99.9) $\frac{E[C_1]}{E[W_1]} = 247.9$	(270,56,76) $\frac{E[C_1]}{E[W_1]} = 300.7$	(280,60,72) $\frac{E[C_1]}{E[W_1]} = 332.2$	(290,52,66) $\frac{E[C_1]}{E[W_1]} = 360.4$	$\lambda = 1$ $\frac{E[C_1]}{E[W_1]} = 248.0$
2	(236,60,99.2) $\frac{E[C_1]}{E[W_1]} = 247.9$	(240,60,99.9) $\frac{E[C_1]}{E[W_1]} = 248.0$	(270,56,76) $\frac{E[C_1]}{E[W_1]} = 300.7$	(280,60,72) $\frac{E[C_1]}{E[W_1]} = 332.2$	$\lambda = 2$ $\frac{E[C_1]}{E[W_1]} = 246.7$
3	(187,56,131) $\frac{E[C_1]}{E[W_1]} = 246.7$	(236,60,99.2) $\frac{E[C_1]}{E[W_1]} = 247.9$	(240,60,99.9) $\frac{E[C_1]}{E[W_1]} = 248.0$	(270,56,76) $\frac{E[C_1]}{E[W_1]} = 300.7$	$\lambda = 1$ $\frac{E[C_1]}{E[W_1]} = 245.9$
4	(172,60,144) $\frac{E[C_1]}{E[W_1]} = 245.9$	(187,56,131) $\frac{E[C_1]}{E[W_1]} = 246.7$	(236,60,99.2) $\frac{E[C_1]}{E[W_1]} = 247.9$	(240,60,99.9) $\frac{E[C_1]}{E[W_1]} = 248.0$	Stop

Table 45.4 The effect of (L_1, L_2) on P_p for a given inspection sequence

L_1	L_2	P_p
200	60	0.5910
190	58	0.5928
180	56	0.5936
170	54	0.5948
160	52	0.5950
150	50	0.5968

Table 45.5 The effect of the inspection sequence on P_p for fixed PM values

I_1	P_p
110	0.642
120	0.610
130	0.578
140	0.510
150	0.480
160	0.430

and $I_1 = 144$. From Table 45.4, we observe that the probability P_p slightly increases as both L_1 and L_2 decrease. This in fact shows that one would perform more PMs than CMs when L_1 and L_2 both become smaller.

Similarly, Table 45.5 presents the probability that the cycle will end due to a PM action, P_p, for various values of I_1 given $L_1 = 172$, $L_2 = 60$ and $\alpha = 0.97$. From Table 45.5, we observe that the probability P_p decreases as I_1 increases. In other words, the maintenance cycle will be more likely to end due to corrective rather than preventive maintenance if one delays in-

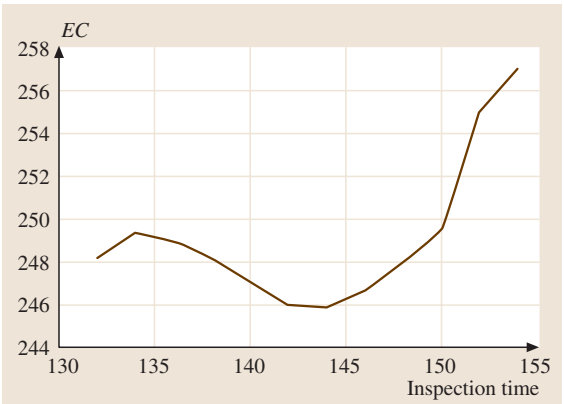


Fig. 45.18 The average maintenance cost $EC(L_1, L_2, I_1)$ versus I_1

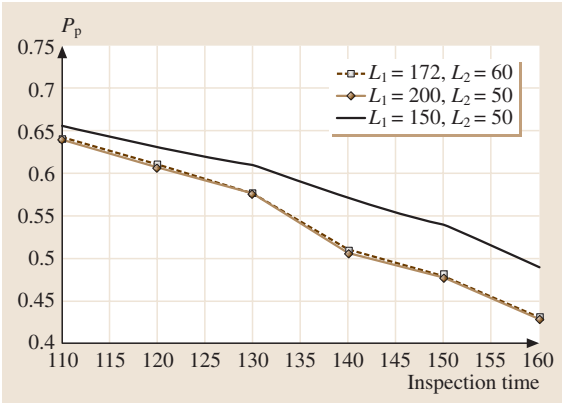


Fig. 45.19 The probability P_p versus I_1 for various pairs (L_1, L_2)

spection. This result can help maintenance managers or inspectors to allocate resources as well as time allocations.

Figure 45.19 shows the results for the probability that the cycle ends due to PM versus the inspection interval time I_1 for given values of the threshold PM levels (L_1, L_2). It is interesting to observe from Fig. 45.19 that the results are about the same for the two combinations of $(L_1 = 172,\ L_2 = 60)$ and $(L_1 = 200,\ L_2 = 50)$.

45.4 Conclusions and Perspectives

In this chapter, we present reliability and maintenance models for degraded systems with multiple competing failure processes. For mathematical modeling, it is always necessary to make some assumptions in order to make the model applicable in practice. For reliability and maintenance modeling, assumptions have often played an important role in determining the structure and complexity of the models.

The results of the maintenance models in this chapter can be used to help practitioners and inspectors as well as marketing managers to allocate resources and for the promotion strategies for new products. It would be of interest for future research to implement these results by collecting data and observing product system degradations in practice. Other research problems worth exploring in the future are as follows [45.22].

1. The objective function discussed in this chapter is to minimize the expected long-run maintenance cost. In practice, costs associated with inspections, preventive maintenance, corrective maintenance, and downtime are sometimes difficult to obtain, even when used in practice. For some critical systems, the overriding goal is to ensure that the system should be available when needed; availability is, therefore, of primary interest, and cost is secondary.
   To achieve a high level of availability for a specified inspection rate, it is worth to determine the optimum number of inspections with respect to imperfect repairs, such as minimal and opportunistic schemes, that maximizes the system availability. The time required for imperfect repairs and for replacement policies are random.
2. This chapter assumed that at any time there is unlimited supply of systems available for replacement. In reality, this assumption might not be true due to budget allocation and other constraints. In this case, a random lead time for delivering the new system when needed should be considered. It is essential and practical to analyze the effect of this random lead time on availability. When incorporating random lead time, the expected downtime will increase; therefore, system availability will decrease.

45.5 Appendix A

Jacobian determinant

Below is a 2×2 *Jacobian* determinant

$$J = \frac{y_1(y_2-W)\left(\frac{y_2}{y_1(y_2-W)} - \frac{y_2(y_1-W)}{y_1^2(y_2-W)}\right)[-d(y_1,y_2)-d_1(y_1,y_2)+d_2(y_1,y_2)]}{y_2(y_1-W)(I_{i+1}-I_i)} + d_3(y_1,y_2)\,,\text{where}$$

$$d(y_1,y_2) = \frac{\left[\left(\frac{y_1-W}{y_1(y_2-W)} - \frac{y_2(y_1-W)}{y_1(y_2-W)^2}\right) y_1(y_2-W)^2 I_{i+1}\, e^{\left(\frac{\ln\left(\frac{y_2(y_1-W)}{y_1(y_2-W)}\right)I_{i+1}}{I_{i+1}-I_i}\right)}\right]}{y_2^2(y_1-W)(I_{i+1}-I_i)},\ y_1 \neq W,\ y_2 \neq W\,,$$

$$d_1(y_1, y_2) = \frac{\mathrm{e}^{\left(\frac{\ln\left(\frac{y_2(y_1-W)}{y_1(y_2-W)}\right)I_{i+1}}{I_{i+1}-I_i}\right)}}{y_2}, \; y_2 \neq W,$$

$$d_2(y_1, y_2) = \frac{\mathrm{e}^{\left(\frac{\ln\left(\frac{y_2(y_1-W)}{y_1(y_2-W)}\right)I_{i+1}}{I_{i+1}-I_i}\right)}(y_2-W)}{y_2^2}, \; y_2 \neq W\,,$$

$$d_3(y_1, y_2) = \frac{d_{31}(y_1, y_2)d_{32}(y_1, y_2)}{y_2^3(y_1-W)^2(I_{i+1}-I_i)^2}, \; y_1 \neq W\,,$$

$$d_{31}(y_1, y_2) = \left(\frac{y_1-W}{y_1(y_2-W)} - \frac{y_2(y_1-W)}{y_1(y_2-W)^2}\right) y_1^2(y_2-W)^3, \; y_2 \neq W\,,$$

$$d_{32}(y_1, y_2) = \left(\frac{y_2}{y_1(y_2-W)} - \frac{y_2(y_1-W)}{y_1^2(y_2-W)}\right) I_{i+1}\,\mathrm{e}^{\left(\frac{\ln\left(\frac{y_2(y_1-W)}{y_1(y_2-W)}\right)I_{i+1}}{I_{i+1}-I_i}\right)}, \; y_2 \neq W\,.$$

45.6 Appendix B

A list of all 18 events:

$$E_1^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[L_1 < Y_1(I_{i+1}) \leq G_1, L_2 < Y_2(I_{i+1}) \leq G_2, D(I_{i+1}) \leq S\right]$$

$$E_2^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[L_1 < Y_1(I_{i+1}) \leq G_1, Y_2(I_{i+1}) \leq L_2, D(I_{i+1}) \leq S\right]$$

$$E_3^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) \leq L_1, L_2 < Y_2(I_{i+1}) \leq G_2, D(I_{i+1}) \leq S\right]$$

$$E_4^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[L_1 < Y_1(I_{i+1}) \leq G_1, Y_2(I_{i+1}) > G_2, D(I_{i+1}) \leq S\right]$$

$$E_5^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[L_1 < Y_1(I_{i+1}) \leq G_1, L_2 < Y_2(I_{i+1}) \leq G_2, D(I_{i+1}) > S\right]$$

$$E_6^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[L_1 < Y_1(I_{i+1}) \leq G_1, Y_2(I_{i+1}) > G_2, D(I_{i+1}) > S\right]$$

$$E_7^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) > G_1, L_2 > Y_2(I_{i+1}), D(I_{i+1}) \leq S\right]$$

$$E_8^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) > G_1, L_2 < Y_2(I_{i+1}) \leq G_2, D(I_{i+1}) \leq S\right]$$

$$E_9^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) > G_1, L_2 > Y_2(I_{i+1}), D(I_{i+1}) > S\right]$$

$$E_{10}^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) > G_1, L_2 < Y_2(I_{i+1}) \leq G_2, D(I_{i+1}) > S\right]$$

$$E_{11}^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) > G_1, Y_2(I_{i+1}) > G_2, D(I_{i+1}) \leq S\right]$$

$$E_{12}^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[Y_1(I_{i+1}) > G_1, Y_2(I_{i+1}) > G_2, D(I_{i+1}) > S\right]$$

$$E_{13}^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right] \cap \left[L_1 < Y_1(I_{i+1}) \leq G_1, L_2 > Y_2(I_{i+1}), D(I_{i+1}) > S\right]$$

$$E_{14}^{(i+1)} = \left[Y_1(I_i) \leq L_1, Y_2(I_i) \leq L_2, D(I_i) \leq S\right]$$

$$\cap\left[L_1 > Y_1(I_{i+1}), L_2 > Y_2(I_{i+1}), D(I_{i+1}) \le S\right]$$

$$E_{15}^{(i+1)} = \left[Y_1(I_i) \le L_1, Y_2(I_i) \le L_2, D(I_i) \le S\right] \cap\left[L_1 > Y_1(I_{i+1}), L_2 > Y_2(I_{i+1}), D(I_{i+1}) > S\right]$$

$$E_{16}^{(i+1)} = \left[Y_1(I_i) \le L_1, Y_2(I_i) \le L_2, D(I_i) \le S\right] \cap\left[L_1 > Y_1(I_{i+1}), L_2 < Y_2(I_{i+1}) \le G_2, D(I_{i+1}) > S\right]$$

$$E_{17}^{(i+1)} = \left[Y_1(I_i) \le L_1, Y_2(I_i) \le L_2, D(I_i) \le S\right] \cap\left[L_1 > Y_1(I_{i+1}), Y_2(I_{i+1}) > G_2, D(I_{i+1}) \le S\right]$$

$$E_{18}^{(i+1)} = \left[Y_1(I_i) \le L_1, Y_2(I_i) \le L_2, D(I_i) \le S\right] \cap\left[Y_1(I_{i+1}) \le L_1, Y_2(I_{i+1}) \le L_2, D(I_{i+1}) \le S\right]$$

References

45.1 J. Tomsky: Regression models for detecting reliability degradation, Proc. Annual Reliability Maintainability Conference , 238–244 (1982)

45.2 W. Nelson: *Accelerated Testing: Statistical Methods, Test Plans, and Data Analysis* (Wiley, New York 1990)

45.3 H. J. Lu: The Use of Degradation Measures In Assessing Reliability. Ph.D. Thesis (Iowa State Univ., Ames, Iowa 1992)

45.4 G. Levitin: Reliability of multi-state systems with two failure-modes, IEEE Trans. Reliab. **52**, 340–348 (2003)

45.5 H. Pham, A. Suprasad, R. B. Misra: Reliability and MTTF prediction of *k*-out-of-*n* complex systems with components subjected to multiple stages of degradation, Int. J. Syst. Sci. **27**(10), 995–1000 (1996)

45.6 H. Pham, A. Suprasad, R. B. Misra: Availability and mean life time prediction of multi-stage degraded system with partial repairs, Reliab. Eng. Syst. Safety **56**, 169–173 (1997)

45.7 R. Bris, E. Chatelet, F. Yalaoui: New method to minimize the preventive maintenance cost of series–parallel systems, Reliab. Eng. Syst. Safety **82**, 247–255 (2003)

45.8 A. Grall, C. Berenguer, L. Dieulle: A condition-based maintenance policy for stochastically deteriorating systems, Reliab. Eng. Syst. Safety **76**, 167–180 (2002)

45.9 A. Grall, L. Dieulle, C. Berenguer, M. Roussignol: Continuous-time predictive-maintenance scheduling for a deteriorating system, IEEE Trans. Reliab. **51**(2), 141–150 (2002)

45.10 A. Chelbi, D. Ait-Kadi: An optimal inspection strategy for randomly failing equipment, Reliab. Eng. Syst. Safety **63**, 127–131 (1999)

45.11 G. A. Klutke, Y. J. Yang: The availability of inspected systems subjected to shocks and graceful degradation, IEEE Trans. Reliab. **44**, 371–374 (2002)

45.12 M. J. Zuo, B. Liu, D. N. P. Murthy: Replacement–repair policy for multi-state deteriorating products under warranty, Eur. J. Oper. Res. **123**, 519–530 (2000)

45.13 H. Pham, M. Xie: A generalized surveillance model with applications to systems safety, IEEE Trans. Syst. Man Cybernetics Pt C **32**, 485–492 (2002)

45.14 J. L. Bogdanoff, F. Kozin: *Probabilistic Models of Cumulative Damage* (Wiley, New York 1985)

45.15 W. Li, H. Pham: Reliability modeling of multi-state degraded systems with multi-competing failures and random shocks, IEEE Trans. Reliab. **54**, 297–303 (2005)

45.16 R. M. Feldman: Optimal replacement with semi-Markov shock models, J. Appl. Probab. **13**, 108–117 (1976)

45.17 M. Ohnishi, H. Kawai, H. Mine: An optimal inspection and replacement policy for a deteriorating system, J. Appl. Probab. **23**, 973–988 (1986)

45.18 C. T. Lam, R. H. Yeh: Optimal maintenance-policies for deteriorating systems under various maintenance strategies, IEEE Trans. Reliab. **43**, 423–430 (1994)

45.19 H. Pham: Cost optimization of a class of noncoherent systems, Math. Comput. Model. **15**(6), 15 (1991)

45.20 L. Dieulle, C. Berenguer, A. Gralland, M. Roussignol: Sequential condition-based maintenance scheduling for a deteriorating system, Eur. J. Oper. Res. **150**, 451–461 (2003)

45.21 W. Li, H. Pham: An inspection–maintenance model for systems with multiple competing processes, IEEE Trans. Reliab. **54**, 318–327 (2005)

45.22 W. Li: Reliability and Maintenance Modeling of Multi-state Degraded Systems with Multiple Competing Failure Processes. Ph.D. Thesis (Dept. Industrial Systems Engineering, Rutgers Univ., Piscataway, New Jersey 2005)

45.23 R. L. Rardin: *Optimization in Operations Research* (Prentice Hall, Piscataway 1998)

46. Statistical Models on Maintenance

This chapter discusses a variety of approaches to performing maintenance. The first section describes the importance of preparing for maintenance correctly, by collecting data on unit lifetimes and estimating the reliability of the units statistically using quantities such as their mean lifetimes, failure rates and failure distributions.

Suppose that the time that the unit has been operational is known (or even just the calendar time since it was first used), and its failure distribution has been estimated statistically. The second section of the chapter shows that the time to failure is approximately given by the reciprocal of the failure rate, and the time before preventive maintenance is required is simply given by the pth percentile point of the failure distribution. Standard replacement policies, such as age replacement, in which a unit undergoes maintenance before it reaches a certain age, and periodic replacement, where the unit undergoes maintenance periodically, are also presented.

Suppose that the failure of a unit can only be recorded at discrete times (so the unit completes a specific number of cycles before failure). In the third section, the age replacement and periodic replacement models from the previous section are converted into discrete models. Three replacement policies in which the unit undergoes maintenance after a specific number of failures, episodes of preventive maintenance or repairs, are also presented. The optimum number of units for a parallel redundant system is derived for when each unit fails according to a failure distribution and fails upon some shock with a certain probability.

Suppose that the unit fails when the total amount of damage caused by shocks has exceeded a certain failure level. The fourth section describes the replacement policy in where the unit undergoes maintenance before failure for a cumulative damage model. The optimum damage level at which the unit should be replaced when it undergoes minimal repair upon failure is also derived analytically.

The last part introduces the repair limit policy, where the unit is replaced instead of being repaired if the repair time is estimated to be more than a certain time limit, as well as the inspection with human error policy, where units are checked periodically and failed units are only detected and replaced upon inspection. Finally, the maintenance of a phased array radar is analyzed as an example of the practical use of maintenance models. Two maintenance models are considered in this case, and policies that minimize the expected cost rates are obtained analytically and numerically.

Although system maintenance is important, performing it without understanding the operational parameters of the system first would probably do more harm than good. Therefore, the first step of maintenance is preparation: we have to collect data on the components used in the system, in order to be able to statistically estimate quantities such as mean lifetimes, failure rates, failure distributions, and so on. Secondly, designers, engineers and managers engaged in maintenance work need to be taught standard maintenance techniques from reliability theory, and how to apply them to real systems. After any necessary training has been performed, and data has been collecting about the system, we construct a maintenance plan, which depends upon the system environment and the resources (monetary and manpower) available. After much trial and error, we establish an maintenance scheme that is optimized for the system in question. This approach allows us to minimize or even eliminate any need for hasty maintenance after a severe system failure.

This chapter summarizes standard maintenance policies that can be applied (statistically and stochastically) to practical models. These policies are largely based upon the author's work [46.1]. If we monitor the age of a unit (in either the calendar time, the operating time or the usage time), it is best to perform maintenance at certain times. On the other hand, if we monitor the number of cycles or system uses, or the amount of total damage, wear or stress incurred by a unit, it is best to perform maintenance after a certain number of cycles or amount of wear, respectively. Moreover, it might be necessary to adopt combinations of these approaches.

In this chapter, we begin (in Sect. 46.1) by estimating the "mean time to failure" using the reciprocal of the failure rate. Then we introduce standard replacement policies, such as "age replacement" and "periodic replacement". Section 46.2 considers discrete versions of the general replacement models derived in Sect. 46.1 (in other words, where the maintenance can only be performed at certain times, which is a more realistic scenario). It also presents three models where replacement occurs after a certain number of events, including the number of system failures, the number of times it has undergone preventive maintenance, or the number of repairs that have been made to the system. Further, we discuss how many units should be allowed to fail in a parallel redundant system before the system is replaced. Section 46.3 introduces maintenance models based on cumulative damage; in other words, those where maintenance occurs after a certain amount of wear, stress, fatigue, corrosion, erosion and garbage. Finally, Sect. 46.4 presents another two useful maintenance models. In the first, known as the "repair limit policy", a failed unit is repaired unless the repair time is too long, in which case it is replaced. Second, we consider the "inspection model", with two types of human error. Finally, we give a practical example – the maintenance of a phased array radar – in which failed elements are either replaced at planned times or when they exceed a managerial number.

46.1 Time-Dependent Maintenance

If the failure of a unit during operation would cause serious damage to the whole system, it is important to know the total amount of time the unit has been in operation and to determine when to replace units or perform maintenance before the failure occurs. This is called *age replacement* or *preventive maintenance* (PM). The optimum age replacement policy (which minimizes the expected cost) and the optimum PM policy (which maximizes the availability are derived in *Barlow* and *Proschan* [46.2] and *Nakagawa* [46.1].

If only the unit's calendar operational time (the period of time since the unit was first used; its "age") is known and its failure is not a relatively minor or inexpensive event, it is necessary to perform PM or replace it periodically. This section describes maintenance policies in which a unit undergoes maintenance according to its total operating time (the total amount of time that the unit has been operational) or the age of the unit.

Suppose that X is a random variable that represents this age or total operating time. We then denote the failure distribution by $F(t) \equiv \Pr(X \le t)$ for $0 \le t < \infty$. Let μ and $h(t)$, respectively, be the finite mean time to failure of X and the failure rate, so $\mu \equiv \int_0^\infty \overline{F}(t)\,\mathrm{d}t$ and $h(t) \equiv f(t)/\overline{F}(t)$ where f is the density of F and $\overline{F} \equiv 1 - F$. Further, $H(t) \equiv \int_0^t h(u)\,\mathrm{d}u$ is called the cumulative hazard function, and is given by the relation

$$\overline{F}(t) = \exp\left[-\int_0^t h(u)\mathrm{d}u\right] = \mathrm{e}^{-H(t)}\,. \qquad (46.1)$$

Thus, $F(t)$, $\overline{F}(t)$, $f(t)$, $h(t)$ and $H(t)$ determine one another. Throughout this paper, we use these notations.

46.1.1 Failure Distribution

The most important statistical parameter is to find the mean time to failure (MTTF) of the unit. This is obtained comparatively easily by collecting data on the unit lifetime. Given the estimated failure distribution $F(t)$ of a unit, then the MTTF is given by $\mu \equiv \int_0^\infty \overline{F}(t)\,\mathrm{d}t$.

Next, since the probability that a unit with age T will fail in an interval $(T, t+T)$ is $[F(t+T)-F(T)]/\overline{F}(T)$, its MTTF (which is called the mean residual life) is

$$\frac{1}{\overline{F}(T)}\int_0^\infty \overline{F}(t+T)\,\mathrm{d}t = \frac{1}{\overline{F}(T)}\int_T^\infty \overline{F}(t)\,\mathrm{d}t \qquad (46.2)$$

which decreases from μ to $1/h(\infty)$ when $F(t)$ is an IFR property [46.2]; in other words when $h(t)$ is increasing. Also, in this case

$$\frac{1}{\overline{F}(T)}\int_T^\infty \overline{F}(t)\,\mathrm{d}t \le \frac{1}{h(T)}\,. \qquad (46.3)$$

Thus, $1/h(T)$ is used as an upper estimator for the MTTF of a unit with age T.

When failures occur in a nonhomogeneous Poisson process, the expected number of failures during $(0\,,\,T]$ is given by $H(T)$ [46.3]. Thus, if T_k corresponds to the time that the expected number of failures is k ($k=1\,,\,2\,,\ldots$), so $H(T_k)=k$, we have

$$\int_{T_{k-1}}^{T_k} h(t)\,\mathrm{d}t = 1 \quad \text{or} \quad \int_0^{T_k} h(t)\,\mathrm{d}t = k$$
$$(k=1\,,\,2\,,\ldots)\,. \qquad (46.4)$$

In particular, when $F(t)$ is IFR,

$$T_k \le \frac{1}{h(T_{k-1})} + T_{k-1} \qquad (46.5)$$

In other words, the time where the expected number of failures from T_{k-1} is 1 is less than $1/h(T_{k-1})$. Note that the equations hold in (46.3) and (46.5) when F is exponential.

From the above discussions, it is possible to estimate that the time to the next failure, if it fails at time t, is about $1/h(t)$.

The simplest way to prevent failure is to make sure that the probability of failure is less than $p(0<p<1)$; in other words to compute a pth percentile point that satisfies $F(T_p)=p$. Then, a unit will undergo replacement at time T_p. Of course, we may consider this replacement to be PM.

Another simple method of replacement is to balance the cost of replacement after failure against that of replacement before failure. A cost c_1 is incurred for each failed unit and c_2 ($<c_1$) is incurred for each operational unit. Then, we have $c_1F(T)=c_2\overline{F}(T)$, so $F(T)=c_2/(c_1+c_2)$. We may compute a $p[=c_2/(c_1+c_2)]$th percentile point for the failure distribution $F(t)$.

46.1.2 Age Replacement

Suppose that the operating record of a unit and its failure distribution $F(t)$ are known, and that its failure rate increases with operating time. In this case, if a unit is replaced by a new one, it is called age replacement. The optimum policy for minimizing the expected cost rate is discussed. If a unit is preventively maintained and becomes as good as new, this is called perfect PM [46.3]. The optimum policy for maximizing the availability is discussed [46.2].

Suppose that a unit is replaced at failure or at a planned time T ($0<T\le\infty$), whichever occurs first. Then the expected cost rate is [46.2]

$$C(T) = \frac{c_1F(T)+c_2\overline{F}(T)}{\int_0^T \overline{F}(t)\,\mathrm{d}t}\,, \qquad (46.6)$$

where c_1 is the cost of replacement at failure and c_2 is the cost of replacement at planned time T, with $c_2<c_1$. If $T=\infty$, then the policy corresponds to replacement upon failure, and the expected cost rate is $C(\infty)=c_1/\mu$.

We find an optimum time T^* which minimizes $C(T)$ in (46.6), provided the failure rate $h(t)$ is strictly increasing, with $h(\infty)\equiv\lim_{t\to\infty}h(t)$. Evidently, since $\lim_{T\to0}C(T)=\infty$, a positive T^* ($0<T^*\le\infty$) must exist. Differentiating $C(T)$ with respect to T and setting it equal to zero, we have

$$h(T)\int_0^T \overline{F}(t)\,\mathrm{d}t - F(T) = \frac{c_2}{c_1-c_2}\,. \qquad (46.7)$$

It is easily to see that the left-hand side of (46.7) strictly increases from 0 to $h(\infty)\mu-1$ because $h(t)$ strictly increases. Thus, if $h(\infty)>c_1/[\mu(c_1-c_2)]$, then there is a finite and unique T^* ($0<T^*<\infty$) which satisfies (46.7), and the expected cost rate is

$$C(T^*) = (c_1-c_2)h(T^*)\,. \qquad (46.8)$$

On the other hand, if $h(\infty)\le c_1/[\mu(c_1-c_2)]$ then $T^*=\infty$; in other words, the unit should be replaced at failure.

Next, the unit is repaired at failure or undergoes PM before failure at planned time T ($0<T\le\infty$), whichever

occurs first. Then, the steady-state availability is

$$A(T) = \frac{\int_0^T \overline{F}(t)\,\mathrm{d}t}{\int_0^T \overline{F}(t)\,\mathrm{d}t + \beta_1 F(T) + \beta_2 \overline{F}(T)}\,, \qquad (46.9)$$

where β_1 is the mean time of repair for a failed unit and β_2 is the mean time of PM for an operational unit at time T with $\beta_1 > \beta_2$. Thus, the policy that maximizes the availability is the same one that minimizes the expected cost $C(T)$ in (46.6), except that c_i is replaced by β_i $(i = 1\,,\,2)$.

46.1.3 Periodic Replacement

If a system consists of many kinds of components and only its age is only known, it would be wise to make planned maintenances at periodic times $kT (k = 1\,,\,2\,,\ldots)$ $(0 < T \le \infty)$. We consider three periodic replacements in which a failed unit is replaced, undergoes minimal repair or remains failed.

A new unit begins to operate at time $t = 0$, and a failed unit is discovered instantly and replaced by a new one. Further, a unit is replaced at the periodic time kT, whatever its age. Let $M(t)$ be a renewal function of $F(t)$, so $M(t) \equiv \sum_{j=1}^{\infty} F^{(j)}(t)$, where $F^{(j)}(t)$ $(j = 1\,,\,2\,,\ldots)$ is the j-fold Stieltjes convolution of $F(t)$ with itself and $F^{(0)}(t) = 1$ for $t \ge 0$. Then, the expected cost rate is [46.2],

$$C_1(T) = \frac{c_1 M(T) + c_2}{T}\,, \qquad (46.10)$$

where c_1 is the cost of replacement at each failure.

We seek an optimum time T^* which minimizes the expected cost rate $C_1(T)$ in (46.10). Differentiating $C_1(T)$ with respect to T and setting it equal to zero implies

$$Tm(T) - \int_0^T m(t)\,\mathrm{d}t = \frac{c_2}{c_1}\,, \qquad (46.11)$$

where $m(t)$ is the renewal density of $M(t)$, so $m(t) \equiv \mathrm{d}M(t)/\mathrm{d}t$. This equation is a necessary condition for a finite T^*, and in this case, the expected cost rate is

$$C_1(T^*) = c_1 m(T^*)\,. \qquad (46.12)$$

Next, a unit is always replaced at times kT, but it is not replaced at failure, and hence it remains a failure for the time interval from its failure to its replacement. Then, the expected cost rate is [46.1]

$$C_2(T) = \frac{c_1 \int_0^T F(t)\,\mathrm{d}t + c_2}{T}\,, \qquad (46.13)$$

where c_1 is the cost for the time elapsed between the failure and the replacement of the unit per unit of time.

Differentiating $C_2(T)$ with respect to T and setting it equal to zero,

$$TF(T) - \int_0^T F(t)\,\mathrm{d}t = \frac{c_2}{c_1}\,. \qquad (46.14)$$

Thus, if $\mu > c_2/c_1$ then an optimum and unique T^* exists which satisfies (46.14), and the expected cost rate is

$$C_2(T^*) = c_1 F(T^*)\,. \qquad (46.15)$$

Finally, a unit undergoes only minimal repair at failure; in other words its failure rate remains undisturbed by minimal repair. Let $H(t)$ be a cumulative hazard function. Then, the expected cost rate is [46.2]

$$C_3(T) = \frac{c_1 H(T) + c_2}{T}\,, \qquad (46.16)$$

where c_1 is the cost of minimal repair at failure. Differentiating $C_3(T)$ with respect to T and setting it equal to zero,

$$Th(T) - \int_0^T h(t)\,\mathrm{d}t = \frac{c_2}{c_1}\,. \qquad (46.17)$$

When the failure rate $h(t)$ is strictly increasing, the left-hand side of (46.17) is also strictly increasing. Thus, if a solution T^* exists to (46.17), it is unique and the expected cost rate is

$$C_3(T^*) = c_1 h(T^*)\,. \qquad (46.18)$$

46.2 Number-Dependent Maintenance

Most maintenance models are based on the continuous time failure distributions shown in Sect. 46.1. However, the time to failure of a unit might be discrete; in other words it can be measured by the number of cycles of some kind (such as the number of rotations) before failure. Units such as switching devices, railroad tracks, ball bearings and airplane tires fall into this category. We would often choose to do this if

the unit is used very frequently. In another case, the exact instant of failure of a unit is not recorded; instead the day or even year in which it occurred is noted.

This section summarizes maintenance models where the maintenance depends on the number of cycles completed by a property of the unit [46.4,5]. First, we convert the standard replacement models from the previous section into discrete time models. Second, we consider the case where a unit is maintained preventively by monitoring the number of occurrences of failure, preventive maintenance or repair. Thirdly, we consider a parallel redundant system where the system is replaced preventively if a specified number of units have failed.

46.2.1 Replacement Policies

Often, an operating unit cannot be replaced at the optimum time for some reason, such as a shortage of spare units, a lack of money or workers, and the inconvenience of having the system out of operation when the unit is replaced. Indeed, some units can only be replaced at idle times: a weekend, month-end or year-end.

This section converts the standard age and periodic replacement models in Sect. 46.1.2 and Sect. 46.1.3 into discrete ones. Units are only replaced at times kT where T $(0 < T < \infty)$ has been specified previously. The other notations we use here are the same ones as those used in Sect. 46.1.

Age Replacement

Only the total operating time of the unit is measured. It is assumed that replacement can occur at times kT $(k = 1, 2, \ldots)$: replacement is only allowed in certain time periods kT. A unit is replaced at time NT or at failure, whichever occurs first, and failures are detected immediately when they occur.

From (46.6), the expected cost rate is given by

$$C(N) = \frac{c_1 F(NT) + c_2 \overline{F}(NT)}{\int_0^{NT} \overline{F}(t)\,\mathrm{d}t} \quad (N = 1, 2, \ldots). \tag{46.19}$$

We want to find an optimum number N^* that minimizes $C(N)$ when the failure rate $h(t)$ is strictly increasing. Forming the inequality $C(N+1) - C(N) \geq 0$, we have

$$\frac{F[(N+1)T] - F(NT)}{\int_{NT}^{(N+1)T} \overline{F}(t)\,\mathrm{d}t} \int_0^{NT} \overline{F}(t)\,\mathrm{d}t - F(NT) \geq \frac{c_2}{c_1 - c_2} \quad (N = 1, 2, \ldots). \tag{46.20}$$

It is easy to see that the left-hand side of (46.20) strictly increases to $h(\infty)\mu - 1$ because $h(t)$ strictly increases. Thus, if $h(\infty) > c_1/[\mu(c_1 - c_2)]$ then a finite and unique minimum N^* $(1 \leq N^* < \infty)$ exists that satisfies (46.20).

On the other hand, if $h(\infty) \leq c_1/[\mu(c_1 - c_2)]$ then $N^* = \infty$; in other words the unit should be replaced at failure.

Table 46.1 shows T_p, T^* and N^* values for a given c_1/c_2 ratio, where $T = 1$, $F(T_p) = c_2/(c_1 + c_2)$, and $F(t) = 1 - \exp(-t/100)^2$. This indicates that T^* and N^* have similar values for a given c_1/c_2. T_p is less than T^* at small c_1/c_2, but T_p becomes a good approximation to T^* for large c_1/c_2. This shows that if the replacement cost at time N^*T is lower that that at time T^*, number-dependent maintenance is more useful that time-dependent maintenance.

Periodic Replacement

Let's assume that the unit is replaced at a time NT and upon each failure, and that failures are detected immediately. From (46.10), the expected cost rate is

$$C_1(N) = \frac{c_1 M(NT) + c_2}{NT} \quad (N = 1, 2, \ldots). \tag{46.21}$$

Forming the inequality $C_1(N+1) - C_1(N) \geq 0$ implies that

$$NM[(N+1)T] - (N+1)M(NT) \geq \frac{c_2}{c_1} \quad (N = 1, 2, \ldots).$$

Next, let's suppose that a unit is replaced at time NT, but this time, if it fails, the unit is not replaced until the next scheduled replacement time. Then, from (46.13), the expected cost rate is

$$C_2(N) = \frac{c_1 \int_0^{NT} F(t)\,\mathrm{d}t + c_2}{NT} \quad (N = 1, 2, \ldots). \tag{46.22}$$

Table 46.1 Optimum T^*, N^* for $T = 1$ and percentile T_p when $F(t) = 1 - \exp(-t/100)^2$

c_1/c_2	T^*	N^*	T_p
2	110	109	64
4	59	59	47
6	46	45	39
10	34	34	31
20	23	23	22
40	16	16	16
60	13	13	13
100	10	10	10

Forming the inequality $C_2(N+1)-C_2(N)\geq 0$ implies that

$$\int_0^{NT}\overline{F}(t)\,\mathrm{d}t - N\int_{NT}^{(N+1)T}\overline{F}(t)\,\mathrm{d}t \geq \frac{c_2}{c_1} \quad (N=1,2,\ldots). \tag{46.23}$$

Since $\overline{F}(t)$ decreases to 0, the left-hand side of (46.23) increases to μ. Thus, if $\mu > c_2/c_1$, a finite and unique minimum N^* $(1\leq N^*<\infty)$ exists which satisfies (46.23).

Finally, a unit is replaced at time NT and undergoes only minimal repair at failures between replacements. Then, from (46.16), the expected cost rate is

$$C_3(N)=\frac{c_1H(NT)+c_2}{NT} \qquad (N=1,2,\ldots). \tag{46.24}$$

Forming the inequality $C_3(N+1)-C_3(N)\geq 0$ implies that

$$NH[(N+1)T]-(N+1)H(NT)\geq \frac{c_2}{c_1} \quad (N=1,2,\ldots). \tag{46.25}$$

When the failure rate $h(t)$ strictly increases to ∞, the left-hand side of (46.25) also strictly increases to ∞. In this case, a finite and unique minimum N^* $(1\leq N^*<\infty)$ exists which satisfies (46.25).

46.2.2 Number-Dependent Replacement

We now consider three replacement models where a unit is replaced after a certain number of events (failures, PMs, repairs).

Number of Failures

Consider the periodic replacement in which a unit is replaced at failure N $(N=1,2,\ldots)$ after its installation, and undergoes only minimal repair upon failure between replacements [46.6, 7]. The notation used is the same as in Sect. 46.1.3. Then, the expected cost rate is

$$C(N)=\frac{(N-1)c_1+c_2}{\sum_{j=0}^{N-1}\int_0^\infty p_j(t)\,\mathrm{d}t} \quad (N=1,2,\ldots), \tag{46.26}$$

where $p_j(t)\equiv\{[H(t)]^j/j!\}\mathrm{e}^{-H(t)}$ $(j=0,1,2,\ldots)$ represents the probability that j failures occur in an interval $(0,t]$.

From the inequality $C(N+1)-C(N)\geq 0$, we have

$$\frac{\sum_{j=0}^{N-1}\int_0^\infty p_j(t)\,\mathrm{d}t}{\int_0^\infty p_N(t)\,\mathrm{d}t}-(N-1)\geq\frac{c_2}{c_1} \quad (N=1,2,\ldots). \tag{46.27}$$

When $h(t)$ strictly increases, the left-hand side of (46.27) also strictly increases, since $\int_0^\infty p_N(t)\,\mathrm{d}t$ decreases to $1/h(\infty)$ [46.7]. Thus, if $h(\infty)>c_2/(\mu c_1)$, a finite and unique minimum N^* exists which satisfies (46.27).

Number of PM Events

Assume that the unit undergoes PM at the planned times kT $(k=1,2,\ldots)$ and its operational age becomes x units of time younger at each PM event, where both x and T $(0\leq x\leq T)$ are constant and have been specified previously. Only minimal repair is performed when the unit fails between replacements. Suppose that the unit is replaced if it operates for the time interval NT. Then, the expected cost rate is (from [46.5])

$$C(N)=\frac{1}{NT}\left[c_1\sum_{j=0}^{N-1}\int_{j(T-x)}^{T+j(T-x)}h(t)\,\mathrm{d}t+(N-1)c_2+c_3\right] \quad (N=1,2,\ldots),$$

where c_1 is the cost of the minimal repair, c_2 is the cost of PM, and c_3 is the cost of replacement, with $c_3>c_2$.

From the inequality $C(N+1)-C(N)\geq 0$, we have

$$N\int_{N(T-x)}^{T+N(T-x)}h(t)\,\mathrm{d}t-\sum_{j=0}^{N-1}\int_{j(T-x)}^{T+j(T-x)}h(t)\,\mathrm{d}t\geq\frac{c_3-c_2}{c_1} \quad (N=1,2,\ldots). \tag{46.28}$$

When $h(t)$ strictly increases, the left-hand side of (46.28) also strictly increases in N. Thus, if a finite N^* which satisfies (46.28) exists, it is unique and it minimizes $C(N)$.

Number of Repairs

Consider a single unit which is repaired upon failure and then returned to operation. It is assumed that the unit begins to operate at time 0 and that it has a failure distribution $F_1(t)$ with a finite mean μ_1, and after the $(j-1)$th repair $(j=2,3,\ldots)$, it has a new distribution $F_j(t)$ with a mean μ_j, which is different and independent from the previous $F_{j-1}(t)$. The jth repair time has the distribution $G_j(t)$ with a mean β_j $(j=1,2,\ldots)$.

A unit is replaced by a new one upon failure N after its installation; in other words, after the completion of the $(N-1)$th repair, the unit is not repaired – it is simply replaced with a new unit. Then, the expected cost rate is, from [46.5]:

$$C(N)=\frac{(N-1)c_1+c_2}{\sum_{j=1}^{N}\mu_j+\sum_{j=1}^{N-1}\beta_j} \quad (N=1\,,2\,,\dots)\,, \tag{46.29}$$

Here c_1 is the cost of each repair and c_2 is the cost of the replacement.

From the inequality $C(N+1)-C(N)\ge 0$, we have

$$\frac{\sum_{j=1}^{N+1}\mu_j+\sum_{j=1}^{N}\beta_j}{\mu_N+\beta_N}-N\ge\frac{c_2}{c_1} \quad (N=1,2,\dots)\,. \tag{46.30}$$

If $\mu_{j+1}+\beta_j>\mu_{j+2}+\beta_{j+1}$ $(j=0\,,1\,,2\,,\dots)$ where $\beta_0\equiv 0$ in other words the mean time of the cycle from one failure to the next decreases with the number of failures – then the optimum number N^* which satisfies (46.30) is unique.

In particular, suppose that $\mu_{j+1}\equiv a^{j-1}\mu$ and $\beta_j\equiv\beta$ $(j=1\,,2\,,\dots\,;0<a<1)$. Then, if $\mu/\beta>(1-a)(c_2/c_1)$, a finite and unique minimum N^* exists which minimizes $C(N)$.

46.2.3 Parallel System

Consider an n-unit parallel redundant system in which the system is replaced if all units have failed. First, we are interested in the number of units that is the most economical [46.8].

Suppose that each unit has an identical failure distribution $F(t)$ with a finite mean μ. Then an n-unit parallel system has a failure distribution of $[F(t)]^n$, and so the mean time to system failure is $\int_0^\infty[1-F(t)^n]\,\mathrm{d}t$. Thus, the expected cost rate is

$$C(n)=\frac{nc_1+c_2}{\int_0^\infty[1-F(t)^n]\,\mathrm{d}t} \qquad (n=1\,,2\,,\dots)\,, \tag{46.31}$$

where c_1 is the cost of one unit and c_2 is the cost of the replacement.

From the inequality $C(n+1)-C(n)\ge 0$, we have

$$\frac{\int_0^\infty[1-F(t)^n]\,\mathrm{d}t}{\int_0^\infty[F(t)^n-F(t)^{n+1}]\,\mathrm{d}t}-n\ge\frac{c_2}{c_1} \quad (n=1\,,2\,,\dots)\,. \tag{46.32}$$

It is easy to see that the left-hand side of (46.32) strictly increases to ∞. A finite and unique minimum n^* exists which satisfies (46.32) and minimizes $C(n)$.

In particular, when $F(t)=1-\mathrm{e}^{-\lambda t}$, n^* is given by a finite and unique minimum such that

$$(n+1)\sum_{j=1}^{n}\frac{1}{j}-n\ge\frac{c_2}{c_1}\,.$$

Next, consider a parallel redundant system with n units in which units fail through shock at a mean interval of θ. It is assumed that the probability that an operating unit fails at shock j is p_j $(j=1\,,2\,,\dots)$, depending on the number of shocks.

The system is replaced preventively before failure if the total number of failed units is $N+1$, $N+2$, …, $n-1$, or it is replaced if all units have failed; otherwise it is left alone. Then, the expected cost rate is [46.8, 9]

$$C(N)=\frac{c_2+(c_1-c_2)\times\sum_{j=1}^{\infty}\sum_{r=0}^{N}\binom{n}{r}[p(j)]^{n-r}[P(j-1)]^r}{\theta\sum_{j=1}^{\infty}j\sum_{m=N+1}^{n}\binom{n}{m}[1-P(j)]^{n-m}\times\sum_{r=0}^{N}\binom{m}{r}[p(j)]^{m-r}[P(j-1)]^r} \quad (N=0\,,1\,,2\,,\dots\,,n-1)\,, \tag{46.33}$$

where c_1 is the cost of replacement for a failed system and c_2 is the cost of replacement for a system before failure with $c_2<c_1$, $P(j)\equiv\sum_{i=1}^{j}p(i)$ $(j=1\,,2\,,\dots)$ and $P(0)\equiv 0$.

If n and p_j are given, we can determine an optimum number N^* that minimizes the expected cost rate $C(N)$ in (46.33) by comparing $N=0\,,1\,,2\,,\dots\,,n-1$. If p_j is a geometric distribution, so $p_j=pq^{j-1}$ $(j=1\,,2\,,\dots\,;q\equiv 1-p,0<p<1)$, then

$$C(N)=\frac{c_2+(c_1-c_2)\sum_{r=0}^{N}\binom{n}{r}(-1)^r p^{n-r}\times\sum_{i=0}^{r}(-1)^i\left[1/(1-q^{n-i})\right]}{\theta\sum_{r=0}^{N}\binom{n}{r}(-1)^r\times\sum_{i=0}^{N-r}\binom{n-r}{i}\left[1/\left(1-q^{n-i}\right)\right]} \quad (N=0\,,1\,,2\,,\dots\,,n-1)\,. \tag{46.34}$$

In particular, when $n=2$,

$$C(0)=\left[c_1p^2+c_2\left(1-p^2-q^2\right)\right]/\theta\,, \tag{46.35}$$

$$C(1) = \frac{c_1}{\theta}\frac{1-q^2}{1+2q} \,. \tag{46.36}$$

Thus, if $c_2/c_1 < q/(1+2q)$, then the system is replaced when one unit fails, and if $c_2/c_1 \geq q/(1+2q)$ then it is replaced when two units have failed. Since $q/(1+2q) < 1/3$, if $c_1 \leq 3c_2$ then the system is replaced when two units have failed.

46.3 Amount-Dependent Maintenance

Some units are maintained preventively by monitoring their amount of wear, stress, fatigue, corrosion, erosion and garbage. A typical model in this case is the "cumulative damage model" or the "shock model", in which a unit fails when the cumulative amount of damage from shocks has exceeded a particular failure level [46.10].

This section discusses the replacement policy for a cumulative damage model where a unit is replaced before failure at damage Z, or upon failure, whichever occurs first. Then we consider the replacement policy for the scenario where a unit undergoes minimal repair upon failure. Optimum damage levels Z^* which minimize the expected cost rates are analytically derived for both replacement policies. The methods and results in this section can be applied to actual units by cmonitoring their deterioration, and using this to decide which form of maintenance is appropriate. If the amount of total damage is proportional to the total operating time and the number of uses, these correspond to the time- and number-dependent maintenance models, respectively.

46.3.1 Replacement Policies

Consider a unit that should operate for an infinite time span. It is assumed that shocks occur at time intervals of X_i and that each shock causes an amount of damage W_i to the unit. The total damage to the unit is additive. A unit fails when the total damage has exceeded a failure level K. Unit failure shoudl be avoided during actual operation if it is costly or dangerous. In such situations, it would be wise to replacement the unit or perform preventive maintenance before failure, since it would be less expensive.

It would be better to adopt the damage level as the trigger for replacement if we know the total damage at any time. We replace a unit before failure when the total damage has exceeded a threshold level Z $(0 \leq Z \leq K)$. That is, we can investigate the total damage immediately after each shock occurs. If the total damage exceeds Z and is less than K, we replace the unit before it fails. If the total damage has exceeded K, it has failed and is replaced; otherwise we leave it alone.

We denote that $F(t) \equiv \Pr(X_i \leq t)$ and $G(x) \equiv \Pr(W_i \leq x)$ $(i = 1, 2, \ldots)$, where $\theta \equiv E(X_i) < \infty$ and $\beta \equiv E(W_i) < \infty$. Therefore, the expected cost rate is (from [46.11]):

$$C(Z) = \frac{c_2}{\theta[1+M(Z)]} + \frac{(c_1-c_2)}{\theta[1+M(Z)]} \times \left\{1 - G(K) + \int_0^Z [1-G(K-x)]\,\mathrm{dM}(x)\right\} , \tag{46.37}$$

where c_1 is the cost of replacement at failure level K, c_2 is the cost of replacement at threshold level Z with $c_2 < c_1$, and $M(x) \equiv \sum_{j=1}^{\infty} G^{(j)}(x)$ represents the number of shocks expected before the total damage exceeds x. It is evident that

$$C(0) = \{c_1[1-G(K)] + c_2 G(K)\}/\theta \,,$$
$$C(\infty) = \frac{c_1}{\theta[1+M(K)]} \,.$$

We find an optimum threshold level Z^* which minimizes the expected cost rate $C(Z)$. Differentiating $C(Z)$ with respect to Z and setting it equal to zero,

$$\int_{K-Z}^{K} [1+M(K-x)]\,\mathrm{d}G(x) = \frac{c_2}{c_1-c_2} \,. \tag{46.38}$$

The left-hand side of (46.38) strictly increases from 0 to $M(K)$. Thus, if $M(K) > c_2/(c_1-c_2)$ then a finite and unique Z^* $(0 < Z^* < K)$ exists which satisfies (46.38), and the expected cost rate is

$$C(Z^*) = (c_1-c_2)[1-G(K-Z^*)]/\theta \,. \tag{46.39}$$

On the other hand, if $M(K) \leq c_2/(c_1-c_2)$ then $Z^* = K$, so the unit should be replaced after failure.

Also, if the unit is replaced upon failure, at damage Z, at time T or upon shock number N (whichever occurs

first), the expected cost rate is

$$C(Z,T,N)=\left(c_2+(c_1-c_2)\sum_{j=0}^{N-1}F^{(j+1)}(T)\times\int_0^Z[1-G(K-x)]\,\mathrm{d}G^{(j)}(x)+(c_3-c_2)\sum_{j=0}^{N-1}\left[F^{(j)}(T)-F^{(j+1)}(T)\right]G^{(j)}(Z)+(c_4-c_2)F^{(N)}(T)G^{(N)}(Z)\right)\times\left(\sum_{j=0}^{N-1}G^{(j)}(Z)\int_0^T\left[F^{(j)}(t)-F^{(j+1)}(t)\right]\mathrm{d}t\right)^{-1},$$

where c_3 is the cost of replacement at time T and c_4 is the cost of replacement at shock N.

46.3.2 Replacement with Minimal Repair

Suppose that a unit fails with probability $p(z)$ upon each shock (z is the total damage), where $p(0)\equiv 0$ and $p(\infty)=1$ [46.12], and that the unit undergoes only minimal repair upon failure. Further, a unit is replaced upon damage Z, at time T or upon shock number N (whichever occurs first). Therefore, the expected cost rate is (from [46.13]):

$$C(Z,T,N)=\frac{c_1\sum_{j=1}^{N-1}F^{(j)}(T)\int_0^Z p(z)\,\mathrm{d}G^{(j)}(z)+c_2\sum_{j=1}^{N}F^{(j)}(T)\left[G^{(j-1)}(Z)-G^{(j)}(Z)\right]+c_3\sum_{j=0}^{N-1}\left[F^{(j)}(T)-F^{(j+1)}(T)\right]G^{(j)}(Z)+c_4F^{(N)}(T)G^{(N)}(Z)}{\sum_{j=0}^{N-1}G^{(j)}(Z)\int_0^T\left[F^{(j)}(t)-F^{(j+1)}(t)\right]\mathrm{d}t},\tag{46.40}$$

where c_1 is the cost of minimal repair upon failure, c_2 is the cost of replacement upon damage Z, c_3 is the cost of replacement at time T, and c_4 is the cost of replacement upon shock N.

The expected cost rate when a unit is only replaced at damage Z is

$$C(Z)\equiv\lim_{\substack{T\to\infty\\N\to\infty}}C(T,N,Z)=\frac{c_1\int_0^Z p(x)\,\mathrm{d}M(x)+c_2}{\theta[1+M(Z)]}.\tag{46.41}$$

Differentiating $C(Z)$ with respect to Z and setting it equal to zero, we have

$$\int_0^Z[1+M(x)]\,\mathrm{d}p(x)=\frac{c_2}{c_1}\tag{46.42}$$

which strictly increases in Z when $p(x)$ strictly increases. Thus, if a solution satisfying (46.42) exists then it is unique.

In particular, when $p(x)=1-\mathrm{e}^{-sx}$ for $s>0$,

$$\int_0^\infty[1+M(x)]\,\mathrm{d}p(x)=\frac{1}{1-G^*(s)},$$

where $G^*(s)$ is the Laplace–Stieltjes transform of $G(x)$. Therefore, if $1/[1-G^*(s)]>c_2/c_1$, then a finite and unique Z^* exists which satisfies (46.42), and the expected cost rate is $c_1p(Z^*)$. On the other hand, if $1/[1-G^*(s)]\le c_2/c_1$ then $Z^*=\infty$ and $C(\infty)=c_1/\theta$.

Similarly,

$$C(T)\equiv\lim_{\substack{N\to\infty\\Z\to\infty}}C(T,N,Z)=\frac{c_1\sum_{j=1}^{\infty}F^{(j)}(T)\int_0^\infty p(z)\,\mathrm{d}G^{(j)}(z)+c_3}{T},\tag{46.43}$$

$$C(N)\equiv\lim_{\substack{T\to\infty\\Z\to\infty}}C(T,N,Z)=\frac{c_1\sum_{j=1}^{N-1}\int_0^\infty p(z)\,\mathrm{d}G^{(j)}(z)+c_4}{\theta N}.\tag{46.44}$$

We can discuss the optimum T^* and N^*, which minimize the expected cost rates $C(T)$ and $C(N)$, respectively.

46.4 Other Maintenance Models

We now introduce two maintenance models that are interesting statistically. One is the *repair limit policy* and the other is *inspection with human errors*. Further, we give an example of the practical application of a maintenance policy, for a phased array radar.

46.4.1 Repair Limit Policy

In the previous sections, we have dealt with replacement and PM policies in which a unit undergoes maintenances at time T or upon failure (whichever occurs first). One alternative, considered here, is to repair a failed unit if the repair time is short but to replace it if the repair time is long. That is, if the estimated repair time of a failed unit is greater than a specified time T, which is called the *repair limit time*, then it is replaced.

It is assumed that the repair time has a general distribution $G(t)$ with a finite mean β. Let c_1 be the replacement cost of the failed unit and $c_r(t)$ be the expected repair cost during $(0, t]$ when the failed unit undergoes repair. Suppose that when the unit fails, its repair time is estimated. If the repair time is estimated to be less than T, it is repaired; otherwise it is replaced. The expected cost rate is, from [46.14],

$$C(T) = \frac{c_1 \overline{G}(T) + \int_0^T c_r(t)\,\mathrm{d}G(t)}{\mu + \int_0^T t\,\mathrm{d}G(t)} . \qquad (46.45)$$

Evidently

$$C(0) = c_1/\mu ,$$
$$C(\infty) = \frac{\int_0^\infty c_r(t)\,dG(t)}{\mu+\beta} .$$

In particular, when $c_r(t) = ct$, the expected cost rate is

$$C(T) = \frac{c_1 \overline{G}(T) + c\int_0^T t\,\mathrm{d}G(t)}{\mu + \int_0^T t\,\mathrm{d}G(t)} . \qquad (46.46)$$

Differentiating $C(T)$ with respect to T and setting it equal to zero, we have

$$\frac{\mu c}{c_1} = \frac{\mu + \int_0^T \overline{G}(t)\,\mathrm{d}t}{T} \qquad (46.47)$$

whose right-hand side strictly decreases from ∞ to 0. Thus, a finite and unique optimum repair limit time T^* exists which satisfies (46.47).

46.4.2 Inspection with Human Errors

Suppose that an operating unit is checked at times kT $(k = 1, 2, \ldots)$ for $0 < T < \infty$, and that failed units are detected only through inspection and are then replaced. Now, two types of human error can occur when the standby unit is checked at periodic times kT $(k = 1, 2, \ldots)$ [46.15]:

1. *Type I human error*: The operational unit is judged to have failed.
2. *Type II human error*: The failed unit is judged to be operational.

It is assumed that the probabilities of type I and type II errors occurring are, respectively, p_1 and p_2, where $0 \le p_1 + p_2 < 1$. In this case, the number of inspections needed to detect and replace a failed unit is

$$\sum_{j=0}^{\infty} j p_2{}^{j-1}(1-p_2) = \frac{1}{1-p_2} .$$

Consider one cycle from time $t = 0$ to the time when a failed unit is detected by perfect inspection or a good unit is replaced in a type I error, whichever occurs first. Let c_1 be the cost of each inspection and c_2 be the cost of the lost operational time elapsed between a failure and its detection per unit of time. Then, the total expected cost of one cycle is

$$\begin{aligned} C(T) = \sum_{j=0}^{\infty}(1-p_1)^j \Bigg\{ \int_{jT}^{(j+1)T} \Bigg[c_1 \left(j + \frac{1}{1-p_2} \right) \\ + c_2 \left(jT + \frac{T}{1-p_2} - t \right) \Bigg] \mathrm{d}F(t) \\ + p_1 c_1 (j+1) \overline{F}((j+1)T) \Bigg\} \\ = (c_1 + c_2 T) \Bigg\{ \frac{1}{1-p_2} \sum_{j=0}^{\infty} (1-p_1)^j \\ \left[\overline{F}(jT) - \overline{F}((j+1)T) \right] \\ + \sum_{j=0}^{\infty} (1-p_1)^j \overline{F}[(j+1)T] \Bigg\} \\ - c_2 \sum_{j=0}^{\infty} (1-p_1)^j \int_{jT}^{(j+1)T} \overline{F}(t)\,\mathrm{d}t . \end{aligned} \qquad (46.48)$$

When $p_1 = p_2 = 0$ (the inspection is perfect), the expected cost is

$$C(T) = (c_1 + c_2 T) \sum_{j=0}^{\infty} \overline{F}(jT) - c_2 \mu \qquad (46.49)$$

which agrees with that of the standard inspection policy [46.1].

In particular, when $F(t) = 1 - \mathrm{e}^{-\lambda t}$, the expected cost can be rewritten as

$$\begin{aligned} C(T) = & (c_1 + c_2 T) \frac{\left(1 - \mathrm{e}^{-\lambda T}\right)/(1-p_2) + \mathrm{e}^{-\lambda T}}{1 - (1-p_1)\mathrm{e}^{-\lambda T}} \\ & - \frac{c_2}{\lambda} \frac{1 - \mathrm{e}^{-\lambda T}}{1 - (1-p_1)\mathrm{e}^{-\lambda T}} . \end{aligned} \qquad (46.50)$$

Differentiating $C(T)$ with respect to T and setting it equal to zero,

$$\frac{e^{\lambda T}-1}{\lambda}[1-p_2(1-p_1)e^{-\lambda T}]-(1-p_1-p_2)T = \frac{c_1}{c_2}(1-p_1-p_2)\,. \tag{46.51}$$

It is evident that the left-hand side of (46.51) strictly increases from 0 to ∞. Therefore, a finite and unique T^* exists which satisfies (46.51).

46.4.3 Phased Array Radar

Finally, we consider an example scenario of the maintenance of a phased array radar (PAR) [46.16]. A PAR consists of a large number of small and homogeneous elements, and it steer the electromagnetic wave used for detection by shifting the signal phases of waves that are radiated from these individual elements [46.17].

Keithely [46.18] showed that the maintenance model applied to a PAR with 1024 elements had a strong influence on its availability. *Heresh* [46.19] discussed the following three maintenance models for a PAR in which all failed elements were detected immediately, calculated the average time to failures of the equipment, and derived its availability:

1. *Immediate maintenance*: Failed elements are detected, localized and replaced immediately.
2. *Cyclic maintenance*: Failed elements are detected, localized and replaced periodically.
3. *Delayed maintenance*: Failed elements are detected and localized periodically, and replaced when they have exceeded a prespecified managerial number.

In real world scenarios, immediate maintenance is rarely adopted because frequent maintenance degrades the availability of the system. Cyclic or delayed maintenance are the most common approaches.

In this section, we investigate the periodic detection of failed elements of the PAR. The PAR consists of N_0 elements, and failures are detected at periodic times kT $(k=1\,,2\,,\dots)$ for a given T $(0<T<\infty)$. If the number of failed elements has exceeded a failure number n $(0<n\le N_0)$, the PAR cannot maintain the required level of radar performance, resulting in operational errors such as target oversight

Cyclic Maintenance

We consider the following cyclic maintenance of the PAR:

1. The PAR consists of N_0 elements which have an identical constant failure rate λ_0. The number of failed elements during $(0\,,t]$ has a binomial distribution with a mean $N_0[1-\exp(-\lambda_0 t)]$. Since N_0 is large and λ_0 is very small, it can be assumed that failures occur approximately according to a Poisson process with a mean $\lambda\equiv N_0\lambda_0$. That is, the probability that j failures occur during $(0\,,t]$ is
$$p_j(t)\equiv\frac{(\lambda t)^j}{j!}e^{-\lambda t}\qquad(j=0\,,1\,,2\,,\dots)\,.$$
2. When the number of failed elements has exceeded a failure number n, the PAR cannot maintain the required level of radar performance.
3. Failed elements are checked at periodic times kT $(k=1,2,\dots)$, and the checking time is negligible.
4. Failed elements are replaced by new ones at time KT $(K=1,2,\dots)$ or at the time when they exceed n, whichever occurs first.

We now introduce some costs. Cost c_1 is the replacement cost of one failed element; c_2 is the cost of the operational loss during replacement, and c_3 is the cost of the degradation of radar performance per unit time. Then, the expected cost to replace the failed element is [46.16]

$$\begin{aligned}&\sum_{j=0}^{n-1}(jc_1+c_2)p_j(KT)+\sum_{k=1}^{K}\sum_{j=0}^{n-1}p_j[(k-1)T]\\&\quad\times\sum_{i=n-j}^{\infty}\left\{[(i+j)c_1+c_2]p_i(T)\right.\\&\quad\left.+c_3\int_{(k-1)T}^{kT}(kT-t)\,\mathrm{d}p_i[t-(k-1)T]\right\}\\&=c_2+c_1\lambda T\sum_{k=0}^{K-1}\sum_{j=0}^{n-1}p_j(kT)+\frac{c_3}{\lambda}\sum_{k=0}^{K-1}\sum_{j=0}^{n-1}p_j(kT)\\&\quad\times\sum_{i=n-j}^{\infty}(i+j-n)p_i(T)\end{aligned}$$

and the mean time to replace the element is

$$\begin{aligned}&\sum_{j=0}^{n-1}(KT)p_j(KT)+\sum_{k=1}^{K}\sum_{j=0}^{n-1}p_j[(k-1)T]\\&\quad\times\sum_{i=n-j}^{\infty}(kT)p_k(T)\\&=T\sum_{k=0}^{K-1}\sum_{j=0}^{n-1}p_j(kT)\,.\end{aligned}$$

Thus, the expected cost is

$$C_1(K) = c_1\lambda + \frac{c_2}{T\sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT)} + \frac{(c_3/\lambda)}{T\sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT)} \times \sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT) \sum_{i=n-j}^{\infty}(i+j-n)p_i(T)\,, \quad (K=1\,,2\,,\ldots)\,. \tag{46.52}$$

We seek an optimum number K^* which minimizes (46.52). From the inequality $C_1(K+1)-C_1(K)\geq 0$, we have

$$\sum_{k=0}^{K-1}\sum_{m=0}^{n-1} p_m(kT)\sum_{j=0}^{n-1} p_j(KT)\sum_{i=n-j}^{\infty}(i+j-n)p_i(T) - \sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT)\sum_{i=n-j}^{\infty}(i+j-n)p_i(T) \geq \frac{\lambda c_2}{c_3}\,, \quad (K=1\,,2\,,\ldots)\,. \tag{46.53}$$

Denoting the left-hand side of (46.53) by $Q_1(K)$, it is clear that $Q_1(K)$ increases to $Q_1(\infty)$ in K. Thus, if $Q_1(\infty) > \lambda c_2/c_3$ then a finite and unique minimum K^* exists which satisfies (46.53).

Delayed Maintenance

We now consider a delayed maintenance model for the PAR, which is similar to the model before, but:

4. Failed elements are replaced by new ones only when they have exceeded a managerial number N ($N \leq n$).

Other assumptions are the same as the ones used for cyclic maintenance.

The expected cost to replace the element is [46.16]

$$\sum_{k=1}^{\infty}\sum_{j=0}^{N-1} p_j[(k-1)T]\sum_{i=N-j}^{n-j+1}[(i+j)c_1+c_2]p_i(T) + \sum_{k=1}^{\infty}\sum_{j=0}^{N-1} p_j[(k-1)T]\sum_{i=n-j}^{\infty}\left\{[(i+j)c_1+c_2] \times p_i(T) + c_3\int_{(k-1)T}^{kT}(kT-t)\,\mathrm{d}p_i[t-(k-1)T]\right\}$$

$$= c_2 + c_1\lambda T\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT) + \frac{c_3}{\lambda}\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT) \times \sum_{i=n-j}^{\infty}(i+j-n)p_i(T)$$

and the mean time to replacement is

$$\sum_{k=1}^{\infty}\sum_{j=0}^{N-1} p_j[(k-1)T]\sum_{i=N-j}^{n-j+1}(kT)p_i(T) + \sum_{k=1}^{\infty}\sum_{j=0}^{N-1} p_j[(k-1)T]\sum_{i=n-j}^{\infty}(kT)p_i(T) = T\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)\,.$$

Thus, the expected cost rate is

$$C_2(N) = c_1\lambda + \frac{c_2}{T\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)} + \frac{(c_3/\lambda)}{T\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)} \times \sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)\sum_{i=n-j}^{\infty}(i+j-n)p_i(T) \quad (N=1\,,2\,,\ldots\,,n)\,. \tag{46.54}$$

We seek an optimum number N^* which minimizes (46.54). From the inequality $C_2(N+1)-C_2(N)\geq 0$, we have

$$\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)\sum_{i=1}^{\infty} i[p_{n+i-N}(T)-p_{n+i-j}(T)] \geq \frac{\lambda c_2}{c_3} \quad (N=1\,,2\,,\ldots\,,n)\,. \tag{46.55}$$

Denoting the left-hand side of (46.55) by $Q_2(N)$, it is clear that $Q_2(N)$ increases to $Q_2(\infty)$ in N. Therefore, if $Q_2(\infty) > \lambda c_2/c_3$ then a finite and unique minimum N^* exists which satisfies (46.55).

We now show a numerical example for when $c_1 = 0$, because it does not affect K^* and N^*. Table 46.2 gives the optimum numbers K^* and N^* and the expected costs $C_1(K^*)$ and $C_2(N^*)$ for $T = 24$, 48, 72, ..., 168 h and $\lambda_0 = 1$, 2, 3, ..., 10×10^{-4} h, when $N_0 = 1000$, $n = 100$ and $c_2 = c_3 = 1$. Table 46.2 indicates that both K^* and N^* decrease when T

Table 46.2 Optimum replacement number K^*, failed element number N^*, and the expected costs $C_1(K^*)$ and $C_2(N^*)$

λ_0	T	K^*	K^*T	$C_1(K^*)$	N^*	$C_2(N^*)$	$C_1(K^*)/C_2(N^*)$
	24	31	744	1.38×10^{-3}	93	1.07×10^{-3}	1.29
	48	15	720	1.41×10^{-3}	89	1.10×10^{-3}	1.28
	72	10	720	1.42×10^{-3}	86	1.13×10^{-3}	1.26
1×10^{-4}	96	7	672	1.49×10^{-3}	83	1.16×10^{-3}	1.28
	120	6	720	1.42×10^{-3}	80	1.18×10^{-3}	1.20
	144	5	720	1.42×10^{-3}	77	1.21×10^{-3}	1.17
	168	4	672	1.49×10^{-3}	74	1.24×10^{-3}	1.20
2×10^{-4}		16	384	2.75×10^{-3}	90	2.19×10^{-3}	1.26
3×10^{-4}		11	264	4.19×10^{-3}	87	3.34×10^{-3}	1.25
4×10^{-4}		8	192	5.41×10^{-3}	85	4.54×10^{-3}	1.19
5×10^{-4}		6	144	6.98×10^{-3}	82	5.77×10^{-3}	1.21
6×10^{-4}	24	5	120	8.36×10^{-3}	80	7.03×10^{-3}	1.19
7×10^{-4}		5	120	1.04×10^{-2}	77	8.34×10^{-3}	1.25
8×10^{-4}		4	96	1.06×10^{-2}	75	9.69×10^{-3}	1.09
9×10^{-4}		3	72	1.39×10^{-2}	73	1.10×10^{-2}	1.26
10×10^{-4}		3	72	1.39×10^{-2}	71	1.26×10^{-2}	1.10

and λ_0 increase. It is interesting that the value of K^*T are approximately 720 h when $\lambda_0 = 1\times10^{-4}$. In this example, $C_1(K^*)$ is always greater than $C_2(N^*)$ and $C_1(K^*)/C_2(N^*) \approx 1.2$. Therefore, in this case it is clear that delayed maintenance is more efficient than cyclic maintenance from an economic point of view.

Up to now, we have only discussed the best policies for minimizing costs, without considering another important factor: the availability of the system. To finish this chapter, we now obtain the availabilities for these two maintenance models. Let T_0 be the time required for checks at times kT $(k = 1, 2, \dots)$ and T_1 be the time required for each replacement that occurs at time kT when the number of failed elements exceeds a failure number n or a managerial number N. Then, in a similar way to the way that the expected costs were derived, we can obtain the availabilities. For cyclic maintenance, the availability is given by

$$A_1(K) = \frac{T\sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT) - (1/\lambda) \times \sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT)\sum_{i=n-j}^{\infty}(i+j-n)p_i(T)}{T_1 + (T+T_0)\sum_{k=0}^{K-1}\sum_{j=0}^{n-1} p_j(kT)}, \quad (K = 1, 2, \dots) \tag{46.56}$$

and for delayed maintenance it is given by

$$A_2(N) = \frac{T\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT) - (1/\lambda) \sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)\sum_{i=n-j}^{\infty}(i+j-n)p_i(T)}{T_1 + (T+T_0)\sum_{k=0}^{\infty}\sum_{j=0}^{N-1} p_j(kT)}, \quad (N = 1, 2, \dots, n). \tag{46.57}$$

References

46.1 T. Nakagawa: Maintenance and optimum policy. In: *Handbook of Reliability Engineering*, ed. by H. Pham (Springer, London 2003)

46.2 R. E. Barlow, F. Proschan: *Mathematical Theory of Reliability* (Wiley, New York 1965)

46.3 T. Nakagawa, M. Kowada: Analysis of a system with minimal repair and its application to replacement policy, Eur. J. Oper. Res. **17**, 176–182 (1983)

46.4 T. Nakagawa: Modified, discrete replacement models, IEEE Trans. Reliab. **R-36**, 243–245 (1987)

46.5 T. Nakagawa: A summary of discrete replacement policies, Eur. J. Oper. Res. **17**, 243–245 (1984)
46.6 H. Morimura: On some preventive maintenance policies for IFR, J. Oper. Res. Soc. Jpn. **12**, 94–124 (1970)
46.7 T. Nakagawa: Generalized models for determining optimal number of minimal repairs before replacement, J. Oper. Res. Soc. Jpn. **24**, 325–337 (1981)
46.8 T. Nakagawa: Optimal number of units for a parallel system, J. Appl. Prob. **21**, 431–436 (1984)
46.9 T. Nakagawa: Replacement problem of a parallel system in random environment, J. Appl. Prob. **16**, 203–205 (1979)
46.10 J. D. Esary, A. W. Marshall, F. Proschan: Shock and wear process, Ann. Prob. **1**, 627–649 (1973)
46.11 T. Nakagawa: On a replacement problem of a cumulative damage model, Oper. Res. Q. **27**, 895–900 (1976)
46.12 S. D. Chikte, S. D. Deshmukh: Preventive maintenance and replacement under additive damage, Nav. Res. Logist. Q. **28**, 36–46 (1981)
46.13 T. Nakagawa, M. Kijima: Replacement policies for a cumulative damage model with minimal repair at failure, IEEE Trans. Reliab. **38**, 581–584 (1989)
46.14 H. Mine, T. Nakagawa: Interval reliability and optimum preventive maintenance policy, IEEE Trans. Reliab. **26**, 131–133 (1977)
46.15 J. J. Coleman, I. J. Abrams: Mathematical model for operational readiness, Oper. Res. **10**, 126–138 (1962)
46.16 T. Nakagawa, K. Ito: Comparison of cyclic and delayed maintenances for a phased array radar, J. Oper. Res. Soc. Jpn. **47**, 51–61 (2004)
46.17 E. Brookner: *Phased-array radars* (Artech House, Boston 1991)
46.18 H. M. Keithley: Maintainability impact on system design of a phased array radar, Annual New York Conf. on Electronic Reliab. **7**, 1–10 (1966)
46.19 A. H. Heresh: Maintainability of phased array radar systems, IEEE Trans. Reliab. **16**, 61–66 (1967)

Part F Applications

Part F Applications in Engineering Statistics

Part F contains eight chapters and focuses on applications in engineering statistics. The first chapter in this part, Chapt. 47, introduces the essential mathematical techniques and financial economic concepts that are used to assess the risks of and deal with asset pricing, and the maximum-entropy approach for calculating an approximate risk-neutral distribution. Chapter 48 concentrates on demand-forecasting problems and their applications in industry. It also reviews various common forecasting methods and discusses models that are used to obtain the optimal stock level for spare parts based on some industrial applications. Chapter 49 introduces various approaches including arithmetic and geometric processes to model sequential data with and without trends as alternative ways to model maintenance problems better. The chapter also introduces repair–replacement models for a deteriorating system based on an arithmetico-geometric process. Chapter 50 focuses on Six Sigma and highlights several methodologies and techniques for product development and service design as well as the core methodologies of Six Sigma. The chapter also includes a real case study on printed circuit boards to illustrate the application of Six Sigma.

Chapter 51 discusses multivariate modeling with copulas and its applications in engineering. The chapter describes the concept and classes of copulas, such as elliptical and Archimedean copulas, and statistical inferences of copula-based connections to multivariate distributions given by the data. Chapter 52 focuses on the application of queuing theory to communication systems. The chapter details theoretical and practical aspects of analyzing the traffic-flow control and load-balancing problems in order to reduce congestion and improve load balancing in modern communication systems. Chapter 53 describes the basic principles of support-vector machines for constructing classification and the development of nonlinear regression prediction models for data modeling using support-vector machine algorithms. Finally, Chapt. 54 focuses on the presentation of various spares-optimization models and the importance of optimal system design. The chapter describes the detailed formulation of cost-effective models for repairable and nonrepairable systems and the solution techniques and algorithms used for obtaining optimal design solutions.

47. Risks and Assets Pricing

This chapter introduces the basic elements of risk and financial assets pricing. Asset pricing is considered in two essential situations, complete and incomplete markets, and the definition and use of a number of essential financial instruments is described. Specifically, stocks (as underlying processes), bonds and derivative products (and in particular call and put European and American options) are considered. The intent of the chapter is neither to cover all the many techniques and approaches that are used in asset pricing, nor to provide a complete introduction to financial asset pricing and financial engineering. Rather, the intent of the chapter is to outline through applications and problems the essential mathematical techniques and financial economic concepts used to assess the value of risky assets. An extensive set of references is also included to direct the motivated reader to further and extensive research in this broad and evolving domain of economic and financial engineering and mathematics that deals with asset pricing. The first part of the chapter (The Introduction and Sect. 47.1) deals with a definition of risk and outlines the basic terminology used in asset pricing. Further, some essential elements of the Arrow–Debreu framework that underlies the fundamental economic approach to asset pricing are introduced. A second part (Sect. 47.2), develops the concepts of risk-neutral pricing, no arbitrage and complete markets. A number of examples are used to demonstrate how we can determine a probability measure to which risk-neutral pricing can be applied to value assets when markets are complete. In this section, a distinction between complete and incomplete markets is also introduced. Sections 47.3, 47.4 and 47.5 provide an introduction to and examples of basic financial approaches and instruments. First, Sect. 47.3, outlines the basic elements of the consumption capital asset-pricing model (with the CAPM stated as a special case). Section 47.4 introduces the basic elements of net present value and bonds, calculating the yield curve as well as the term structure of interest rates and provides a brief discussion of default and rated bonds. Section 47.5 is a traditional approach to pricing of options using the risk-neutral approach (for complete markets). European and American options are considered and priced by using a number of examples. The Black–Scholes model is introduced and solved, and extensions to option pricing with stochastic volatility, underlying stock prices with jumps as well as options on bonds are introduced and solved for specific examples. The last section of the chapter focuses on incomplete markets and an outline of techniques that are used in pricing assets when markets are incomplete. In particular, the following problems are considered: the pricing of rated bonds (whether they are default-prone or not), engineered risk-neutral pricing (based on data regarding options or other derivatives) and finally we also introduce the maximum-entropy approach for calculating an approximate risk-neutral distribution.

Risk results from the direct and indirect adverse consequences of outcomes and events that were not accounted for or that we were ill prepared for, and concerns their effects on individuals, firms, financial markets or society at large. It can result from many reasons, internally, externally and strategically induced or resulting from risk externality – namely, when all costs or benefits are not incorporated by the market in the price of the asset, the product or the service received. A definition of risk involves several factors including: (i) consequences, (ii) their probabilities and their distribution, (iii) individual preferences, (iv) collective preferences and (v) sharing, contracts or risk-transfer mechanisms. These are relevant to a broad number of fields as well, each providing a different approach to the measurement, the valuation and the management of risk which *is motivated by psychological needs and the need to deal with problems that result from uncertainty and the adverse consequences they may induce* [47.1, 2]. In this chapter we are primarily concerned with risk and pricing and specifically financial assets pricing.

The definition of risk, risk measurement, risk pricing and risk management are intimately related, one feeding the other to determine the proper levels of risks that an individual seeks to sustain and the market's intended price [47.3–6]. Financial asset pricing has sought primarily to determine approaches and mechanisms for market pricing of these risks while financial risk management and engineering are concerned with the management of financial risks, seeking on the one hand to price private risks and on the other responding to the managerial finance considerations that these risks entail. Financial risk management, for example, deals extensively with hedging problems in order to reduce the risk of a particular portfolio through a trade or a series of trades, or contractual agreements reached to share and induce an efficient risk allocation by the parties involved [47.1, 7–12]. To do so, a broad set of financial instruments were developed, including bonds of various denominations, options of various types etc., in some cases broadly traded, thereby allowing a market mechanism to price these risks. For example, by a judicious use of options, contracts, swaps, insurance and investment portfolios etc. risks can be brought to bearable economic costs and shared by the parties involved in market transactions. These tools are not costless however, and require a careful balancing of the numerous factors that affect risk, the costs of applying these tools and a specification of tolerable risk. For example, options require that a premium be paid to limit the size of losses just as the insured are required to pay a premium to buy an insurance contract to protect them in case of unforeseen accidents, theft, diseases, unemployment, fire, etc. For this reason, private tools such as portfolio investment strategies, value at risk ([47.13–18] based on a quantile risk measurement providing an estimate of risk exposure) are used to manage individual risks. Financial engineering in particular has devoted a substantial attention to reconciling the management of individually priced risks and market-priced risks such that risks can be managed more efficiently. These concerns also reflect the basic approach of finance and the use of financial instruments, currently available through brokers, mutual funds, financial institutions, commodity and stock derivatives etc., which are motivated by three essential reasons [47.19–25]:

- To price the multiplicity of claims, accounting for risks and dealing with the adverse effects of uncertainty or risk (that can be completely unpredictable, partly or wholly predictable)
- To explain and account for investors' behavior. To counteract the effects of regulation and taxes by firms and individual investors (who use a wide variety of financial instruments to bypass regulations and

increase the amount of money investors can make while reducing the risk they sustain).
- To provide a rational framework for individuals' and firms' decision-making and to suit investors needs in terms of the risks they are willing to assume and pay for.

Financial instruments deal with uncertainty and the management of the risks they imply in many different ways. Some instruments merely transfer risk from one period to another and in this sense they reckon with the time phasing of events. One of the more important aspects of such instruments is to supply *immediacy* – i. e. the ability not to wait for a payment. Other instruments provide *spatial diversification* (in other words the distribution of risks across independent investments, classes or geography) and *liquidity*. By liquidity, we mean the cost to convert instantly an asset into cash at its fair price. This liquidity is affected both by the existence of a market (in other words, buyers and sellers) as well as the cost of transactions associated with the conversion of the asset into cash. As a result, essential financial risks include: (a) market-industry specific risks and (b) term structure – currency–liquidity risks. Throughout these problems financial engineering provides a comprehensive set of approaches, techniques and tools that seek to bridge the gap between theory and practice, between individual preferences and market pricing and seeks to provide numerical and computer-aided techniques that respond to the needs of individual investors and financial institutions. Further, it recognizes the centrality of money in decision-making processes: making money, not losing it and protecting investors from adverse consequences. To do so, asset pricing (valuation), forecasting, speculating and risk reduction through fundamental analysis, trading (hedging) are essential activities of traders, bankers and investors alike. Financial engineering, for example, deals extensively with the construction of portfolios, consisting of assets of broadly defined risk–return characteristics, derivatives assets etc. with risk profiles desired by individual investors [47.20, 26–38]. For these reasons, risk and financial engineering are applied not only to financial decision-making. Applications to engineering project valuation, management science, engineering risk economics and so on, are a clear indication of the maturity and the usefulness of financial asset pricing, financial engineering and financial risk management. The purpose of this chapter is to outline and explain the salient factors of these continually renewed and expanding fields of research and applications of asset pricing. At the same time we shall seek to bridge the gap between theory and practice while maintaining a mathematical level accessible to typical finance, risk, management and engineering students familiar with basic notions in probability and stochastic processes.

47.1 Risk and Asset Pricing

Asset pricing, broadly, seeks to reduce assets to the identification of *state prices*, a notion that Arrow has coined, and from which any security has an implied value as the weighted sum of its cash flows, state by state, time by time, with weights given by the associated state prices [47.19, 38, 39]. Such state prices may therefore be viewed as the marginal rates of substitution among state-time consumption opportunities, for an unconstrained investor, with respect to a numeraire good [47.23]. We shall focus first on complete markets, where state prices that make it possible to uniquely value assets do exist. Issues and topics relating to incomplete markets, stochastic volatility etc., are discussed as well. However, this is far too important and far too broad a field to be treated in this chapter's space. We begin by outlining some key terms commonly used in the language of asset pricing. A more explicit and detailed outline of such terms can be found in *Duffie* [47.22, 23], *Karatzas* and *Shreve* [47.40, 41]. We shall also outline the basic ideas of the Arrow–Debreu framework which underlies asset pricing.

47.1.1 Key Terms

In most financial models of asset pricing we use a set of states Ω, with associated probabilities to characterize the model underlying uncertainty. Such a set may be finite or infinite. A set of events is then expressed by $\mathfrak{I}$, also called a tribe and by some a σ-algebra. $\mathfrak{I}$ is a collection of subsets of Ω that can be assigned a probability $P(A)$ denoting the probability of a specific event A. In an intertemporal framework defined by the dates 0, 1, . . . , T+1, there is at each date a tribe $\mathfrak{I}_t \subset \mathfrak{I}$, corresponding to the events based on the information available at that time t. Any event in $\mathfrak{I}_t$ is known at time t to be true or false. The convention $\mathfrak{I}_t \subset \mathfrak{I}_s$, $t \leq s$ is used at all times, meaning

that events are never *forgotten* and therefore information accessed over time provides ever expanding *knowledge*. For simplicity, we let events in $\Im_0$ have probability 0 or 1, meaning that there is no information at time t=0. A filtration is defined by $\Phi = \{\Im_0, \Im_1, \Im_2, \ldots, \Im_T\}$, sometimes called an information structure, representing how information is revealed over time. For any random variable Y, we thus use at time t, $E_t(Y) = E(Y|\Im_t)$, to denote the conditional expectation of Y given $\Im_t$. For notational simplicity, we also let $Y = Z$ for any two random variables Y and Z, if the probability that $Y \neq Z$ is zero (for a review of essential elements in probability see for example, [47.42–48]).

An adapted process, defined by a sequence $X = \{X_0, X_1, X_2, \ldots, X_T\}$ such that, for each t, X_t is a random variable with respect to $(\Omega, \Im_t)$ means, informally, that X_t is observable at time t. An important characteristic of such processes in asset pricing is that an adapted process X is a martingale if, for any times t and $s > t$, we have $E_t(X_s) = X_t$. For example, if the conditional expectation of an asset price equals the currently observed price, then the adapted price process is a martingale. For this reason, important facets of financial asset pricing revolve around the notion of martingales. Another term of importance we use with respect to stochastic price processes is *non-anticipating*. This means that, for any time $t < s$ the function (price) is statistically independent of the *future uncertainty*, or the current price is independent of the future Wiener process $W(s) - W(t)$. These mathematical properties are extremely useful in proving basic results in the theoretical analysis of financial markets. However, in practice, underlying processes might not be martingales and further be anticipative processes. Of course, this will also imply a temporal dependence and our theoretical and financial constructs may be violated.

A security is a claim to an adapted (and non-anticipating) dividend process, say D, with D_t denoting the dividend paid by the security at time t. Each security has an adapted security-price process S, so that S_t is the price of the security, ex dividend, at time t. That is, at each time t, the security pays its dividend D_t and is then available for trade at price S_t. This convention implies that D_0 plays no role in determining ex-dividend prices. The cum-dividend security price at time t is $S_t + D_t$. A trading strategy is an adapted process $\boldsymbol{n}$ in $\mathbb{R}^N$. Here, n_t represents the portfolio held after trading at time t. The dividend process $D^{\boldsymbol{n}}$ generated by a trading strategy $\boldsymbol{n}$ is thus defined by $D_t^{\boldsymbol{n}} = n_{t-1}(S_{t-1} + D_t) - n_t S_t$ with n_{-1} taken to be zero by convention. Consider a portfolio that invests wealth in the security and in a bond B_{t-1}; and say that at time $t-1$, the portfolio wealth state is given by:

$$X_{t-1}^{\boldsymbol{n},m} = n_{t-1} S_{t-1} + m_{t-1} B_{t-1} .$$

We then state that a strategy is said to be self-financing if:

$$X_t^{\boldsymbol{n},m} - X_{t-1}^{\boldsymbol{n},m} = n_{t-1}\,(S_t - S_{t-1}) + m_{t-1}\,(B_t - B_{t-1}) .$$

For example, a bonds-only strategy is defined by $n_{t-1} = 0$, while buy-and-hold (long) strategies (that do not depend on time) imply that $n_{t-1} = n > 0$. A short position is defined when $n_{t-1} < 0$. Finally, a strategy consisting of maintaining a constant proportion of our wealth in bonds and stock means that: $n_t S_t / X_t^{\boldsymbol{n},m} = \alpha$ while $m_t B_t / X_t^{\boldsymbol{n},m} = 1 - \alpha$.

The important notion of arbitrage in asset pricing is used to define a trading strategy that costs nothing to form, never generates losses, and, with positive probability, will produce strictly positive gains at some time. The notion of efficient markets in particular presumes that, for efficient markets to exist, there must not be arbitrage trading strategies. The search for profits by traders and investors is therefore motivated explicitly by the search for arbitrage opportunities. This is of course a rational investment approach, for in the presence of an arbitrage, any rational investor who prefers to increase his dividends would undertake such arbitrage without limit, so markets could not be in equilibrium (in a sense that we shall see later on). As a result, the notion of no arbitrage, and the associated concepts of martingales, risk-neutral pricing and complete markets are fundamental key terms that must be understood to appreciate the scope and the spirit of asset pricing, financial engineering and financial risk management. Further, we shall also distinguish clearly between individual and collective (market) valuation. In the former, the agent-investor is assumed to optimize an expected utility of consumption, subject to an endowment constraint, while in the latter case, that investor will be fully aware of the market valuation of risk, based on an equilibrium in state prices in order to tailor an appropriate and fitting strategy to his preferences. An essential objective of dynamic asset pricing, which deals with the intertemporal and risk effects of asset pricing is to link the collective (multi-agent) equilibrium valuation (pricing) of assets to macroeconomic variables, hopefully, observable. This latter field of study requires an extensive familiarity with economic theory, finance and stochastic calculus.

47.1.2 The Arrow–Debreu Framework

The Arrow–Debreu framework underlies the approach to asset pricing. It is therefore useful to present it, even if briefly (see also [47.19,22,49,50]). Assume that there are N securities, $S_1 \dots S_N$, each of which can be held long or short in a portfolio consisting of these securities. Let $n_i > 0$ be the number of securities S_i currently priced at p_i. Thus, the vectorial product $\boldsymbol{n} \cdot \boldsymbol{p}, \boldsymbol{n} \cdot \boldsymbol{p} = \sum_{i=1}^{N} n_i p_i$ denotes the value (price) of the portfolio held. To each security i, there are associated potential cash flows D_{ij}, $j = 1, 2, \dots, M$ where M is the number of all possible *states* of the market at the end of the trading period. For example, if over one period, the market can assume only two states (say high and low) then the market is binomial and $M = 2$. If it assumes three potential states, then $M = 3$ and the market is trinomial etc. For the portfolio as a whole, we thus have the cash flow matrix:

$$\boldsymbol{nD}_{\cdot j} = \sum_{i=1}^{N} n_i D_{ij},$$
$$\boldsymbol{D} = (D_{ij}), \quad i = 1, \dots, n; \quad j = 1, \dots, M.$$

Where $\boldsymbol{D}_{\cdot j}$ denotes the vector of cash flows for all securities held if state j occurs, while the j-th row of the matrix $\boldsymbol{D}$ represents all possible cash flows associated with holding one unit of the j-th security, including dividend payment and market profit/losses (in dollars). If $n_i > 0$, the investor is long and the investor collects $n_i D_{ij}$ at the end of the period. If $n_i < 0$, the investor is short and the investor has a liability at the end of the period (taken by borrowing securities and selling them at the market price). Further, we assume that the transaction costs, commissions, taxes, etc. are neglected. The cash flow of the portfolio at the j-th state is thus, $\boldsymbol{nD}_{\cdot j}, \forall j \in [1, \dots M]$ as stated above. The cash flow thus defined allows us a formal definition of arbitrage.

Definition (Arbitrage)
An arbitrage portfolio is a portfolio $\boldsymbol{n}$ such that

1. Either $\boldsymbol{nP} = 0$, $\boldsymbol{nD}_{\cdot j} \geq 0$, $\forall j \in [1, \dots M]$ and $\boldsymbol{nD}_{\cdot j} > 0$ for some $j \in [1, \dots M]$,
2. or $\boldsymbol{nP} \leq 0$ and $\boldsymbol{nD}_{\cdot j} \geq 0$ $\forall j \in [1, \dots M]$.

That is, an arbitrage portfolio is a position that either has zero initial cost or has no *downside* regardless of the market outcome, and thus offers the possibility to make money without investment and finally can realize an immediate profit that has no downside.

Inversely, there is no arbitrage if an arbitrage portfolio cannot be constructed. The implication of no arbitrage has a direct implication on asset pricing and to the definition of risk-neutral probabilities which are used to define a price linearly in terms of the markets' cash flows. This is summarized by the following Theorem 47.1 whose proof can be found in *Duffie* [47.23]:

Theorem 47.1 (The fundamental theorem of asset pricing with no arbitrage)
If there exist a vector of positive numbers (also called asset prices) $\pi_j (j = 1 \dots M)$ such that:

$$P_i = \sum_{j=1}^{M} D_{ij} \pi_j, \quad j = 1 \dots M$$

or in vector notation $\boldsymbol{P} \equiv \boldsymbol{D\pi}$

Then there exist no arbitrage portfolios. And conversely, if there are no arbitrage portfolios, there exists a vector $\boldsymbol{\pi}$ with positive entries satisfying $\boldsymbol{P} \equiv \boldsymbol{D\pi}$ [47.23,50].

In this framework, risk-neutral probabilities or equivalently, risk-adjusted probabilities, are defined by

$$\hat{\pi}_j = \frac{\pi_j}{\sum_{j=1}^{M} \pi_k}, \quad j = 1, \dots M, \quad \hat{\pi}_j \geq 0,$$
$$\sum_{j=1}^{M} \hat{\pi}_j = 1.$$

Using these probabilities we can write, based on Theorem 47.1, that an asset price is equal to the discounted value of future cash-flow payments, at a risk-less discount rate R_f, or:

$$P_i = \frac{1}{1 + R_f} \sum_{j=1}^{M} D_{ij} \hat{\pi}_j.$$

To see that this is the case, suppose that there exists an investment opportunity (a pure, risk-free zero bond or a money market deposit) which guarantees (for sure) \$1 at the end of the period. The payoff of a bond is thus $1 \leq j \leq M$, $(1, 1, \dots 1)$ (for all states R in $\mathbb{R}^M$) and, according to Theorem 47.1, $P_{\text{BOND}} = \sum_{j=1}^{M} \pi_j = 1/(1 + R_f)$. Thus, $1 + R_f = 1/\sum_{j=1}^{M} \pi_j$, and R_f is called the bond (in this case risk-free) yield. This can be written as $P_i = \frac{1}{1+R_f} \sum_{j=1}^{M} D_{ij} \hat{\pi}_j$ as stated above, which is the expected value under the risk-adjusted (risk-neutral)

probabilities, which we denote by $E_{\mathrm{RN}}\{\cdot\}$ (not to be confused with the historical distribution of prices). Namely, $E_{\mathrm{RN}}\{\cdot\}$ denotes the expectation associated with the operator associated with probabilities $\hat{\pi}_j$, $1 \le j \le M$. The implications of no arbitrage are summarized by the following Theorem 47.2.

Theorem 47.2 (The fundamental theorem of risk-neutral pricing)
Assume that the market admits no arbitrage portfolios and that there exists a risk-less lending/borrowing at rate R_{f}. Then, there exists a probability measure (risk neutral) defined on the set of feasible market outcomes, $\{1, 2, \ldots M\}$, such that the value of any security is equal to the expected value of its future cash flows discounted at the risk-less lending rate.

To calculate these risk-neutral probabilities, we use a portfolio replication which is defined as follows. Given a security S, and a set of securities $S_1, \ldots S_k$, we say that the portfolio $(n_1, n_2, \ldots n_N)$ *replicates* S if the security and the portfolio have identical cash flows. Further, given two identical cash flows, their price is, necessarily the same, as otherwise there would be an opportunity for arbitrage. This is also called the law of the single price. On the basis of the current analysis, we turn at last to a formal definition of complete markets.

Definition (Complete markets)
A securities market with M states is said to be *complete* if, for any vector cash-flow $(D_{.1} \ldots D_{.M})$, there exists a portfolio of traded securities $(n_1, n_2, \ldots n_N)$ which has cash flow D_j in state $j \in [1, \ldots M]$. Thus market completeness implies that:

$$\boldsymbol{nD} \equiv \boldsymbol{D}, \quad \text{or} \quad \sum n_i D_{ij} = D_j\,,$$
$$j \in [1, \ldots M] \quad \text{has a solution} \quad \boldsymbol{n} \in \mathbb{R}^N$$
$$\text{for any} \quad D \in \mathbb{R}^M\,.$$

This is equivalent to the condition: rank $\boldsymbol{D} \equiv M$.

The implication of this definition is that, if a portfolio can be replicated uniquely (the rank condition $\boldsymbol{D} \equiv M$, providing a unique solution to the linear pricing equations) then there is one price and complete markets can exist. Inversely, the uniqueness of asset prices determines a complete market. This is summarized by the proposition below.

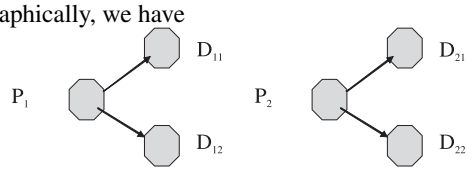
Proposition
Suppose that the market is complete. Then there is a unique set of state prices $(\pi_1 \ldots \pi_M)$ and hence a unique set of risk-neutral probabilities $(\hat{\pi}_1 \ldots \hat{\pi}_M)$. Conversely, if there is a unique set of state prices, then the market is complete, with risk-neutral probabilities $(\hat{\pi}_1 \ldots \hat{\pi}_M)$.

We shall find that the generality of these results is maintained in inter-temporal (dynamic) asset pricing. We consider first some simple examples to demonstrate the pricing implications of the Arrow–Debreu framework.

Example 47.1: Assume a binomial model for a security that assumes two possible future outcomes $\boldsymbol{D} = \{D_1, D_2\}$, $M = 2$. Consider as well two securities with current price P_i, $i = 1, 2$ (for example, a portfolio consisting of a stock and and a bond and a call option on the security with P_1 known and P_2 to be priced), each of these is generating a cash flow (D_{i1}, D_{i2}), $i = 1, 2$. We have no arbitrage if there are risk-neutral probabilities such that, for each security we have

$$P_i = \frac{1}{1+R_{\mathrm{f}}}\left(\hat{\pi}_1 D_{i1} + \hat{\pi}_2 D_{i2}\right), \quad i = 1, 2\,.$$

Graphically, we have



Since $\hat{\pi}_1 + \hat{\pi}_2 = 1$, both the risk-neutral probabilities are given by solving the system of equations in two unknowns for a unique solution (and therefore markets are complete):

$$P_1 = \frac{1}{1+R}\left[\hat{\pi}_1 D_{11} + (1-\hat{\pi}_1) D_{12}\right];$$
$$P_2 = \frac{1}{1+R}\left[\hat{\pi}_1 D_{21} + (1-\hat{\pi}_1) D_{22}\right],$$

where $\hat{\pi}_1$ and P_2 are to be calculated by a solution of these two equations, as:

$$\hat{\pi}_1 = \frac{(1+R)P_1 - D_{12}}{D_{11} - D_{12}}\,,$$
$$P_2 = \frac{P_1}{D_{11} - D_{12}}\,(D_{21} - D_{22}) + \frac{1}{1+R}\left(\frac{D_{11}D_{22} - D_{12}D_{21}}{D_{11} - D_{12}}\right).$$

Equivalently, the matrix $\boldsymbol{D}$ has to be of full rank 2. However, if both assets assume three potential states,

leading to a trinomial model, then of course $M = 3$, while rank $(\boldsymbol{D}) = 2$ and therefore the market is not complete since the number of solutions to this system of equations is infinite. If we add a third asset whose price is based on the same events we obtain a 3×3 matrix with rank 3 whose solution is again unique and therefore the market is complete again. In other words, adding another asset has made it possible to obtain the risk-neutral probabilities, needed for asset pricing and *complete* the market. Similar examples can be used to price simple models of forward futures contracts as well as prove some fundamental equalities in options' finance (put–call parity for example, which we will see subsequently). The Arrow–Debreu framework underlies the basic approach of modern financial economics for asset pricing and therefore it is important to appreciate its basic assumptions as stated here.

47.2 Rational Expectations, Risk-Neutral Pricing and Asset Pricing

Rational (risk neutral) expectations, risk-neutral pricing, complete and incomplete markets, as shown in the Arrow–Debreu framework, underlie the valuation of risk and the use of financial engineering for asset pricing. Rational expectations imply that current prices reflect future uncertainties and their price, and also mean that current prices are based on the unbiased, minimum-variance mean estimate of future prices. This property provides the means to value assets and securities, although in this approach, bubbles are not possible, since they seem to imply a persistent error or bias in forecasting. Rational-expectation pricing will not allow investors to earn above-average returns without taking above-average risks. In such circumstances, arbitrageurs, those *smart investors* who seek to identify returns that have no risk and yet provide a return, will not be able to profit without assuming risks.

The concept of rational expectation is due to *Muth* [47.51], however, who formulated it as a decision-making hypothesis in which agents are informed, constructing a model of the economic environment and using all the relevant and appropriate information at a time a decision is made (see also [47.52], p. 23):

> *I would like to suggest that expectations, since they are informed predictions of future events, are essentially the same as the predictions of the relevant economic theory . . . We call such expectations "rational" . . . This hypothesis can be rephrased a little more precisely as follow: that expectations . . . (or more generally, the subjective probability distribution of outcomes) tend to be distributed, for the same information set, about the prediction of the theory (the objective probability of outcomes).*

In other words, if investors are "smart" and base their decisions on informed and calculated predictions, then, prices equal their discounted expectations. In other words, it implies that investors' subjective beliefs are the same as those of the real world – they are neither pessimistic nor optimistic. When this is the case and a *rational expectations equilibrium* holds, we say that markets are complete or efficient. Samuelson pointed out this notion in 1965 as the martingale property leading *Fama*, *Lucas* and *Harrison* and *Kreps* [47.53–56] to characterize such properties as market efficiency.

Lucas used a concept of rational expectations similar to Muth to confirm Milton Friedman's 1968 hypothesis of the long-run neutrality of monetary policy. Specifically, *Lucas* [47.54] and *Sargent* [47.57] have shown that economic agents alter both their expectations and their decisions to neutralize the effects of monetary policy.

Martingale and the concept of market efficiency are intimately connected, as shown in the Arrow–Debreu framework and pointed out by *Harrison* and *Pliska* [47.39] in their seminal paper. If prices follow a martingale, then only the information available today is relevant to make a prediction on future prices. In other words, the present price has all the relevant information embedding investors' expectations. This means that in practice (the weak form of efficiency) past prices should be of no help in predicting present prices or equivalently prices have no memory. Similarly, if prices follow a martingale and are unpredictable, markets are efficient. In this case, arbitrage is not possible and there is always a party to take on a risk irrespective of how high it is. Hence, risk can be perfectly diversified away and made to disappear. In such a world without risk, all assets behave as if they are risk-free and therefore prices can be discounted at a risk-free rate. This property, justifies our use of risk-neutral pricing (RNP). It breaks down however if any of the previous hypotheses (martingale, rationality, no arbitrage, absence of transaction costs etc.) are invalid. In such a case, prices can no longer be unique and markets are said to be incomplete.

There is a confrontation between economists however, some of whom believe that markets are efficient and some who do not. Obviously, market efficiency fails

to account for market anomalies such as bubbles and bursts, firms' performance and their relationship to size etc. As a result, behavioral finance has sought to provide an alternative dogma (based on psychology) to explain the behavior of financial markets. Whether these dogmas will converge back together as classical and Keynesian economics have, remains yet to be seen. In summary, however, some believe that the current price imbeds all future information, and some presume that past prices and behavior can be used to predict future prices. If the test is to make money, then the verdict is far from reached. Richard Roll, a financial economist and money manager argues:

> *I have personally tried to invest money, my clients and my own, in every single anomaly and predictive result that academics have dreamed up. And I have yet to make a nickel on these supposed market inefficiencies. An inefficiency ought to be an exploitable opportunity. If there is nothing that investors can exploit in a systematic way, time in and time out, then it is very hard to say that information is not being properly incorporated into stock prices. Real money investment strategies do not produce the results that academic papers say they should . . . . . but there are some exceptions including long term performers that have over the years systematically beat the market (Burton Malkiel, The Wall Street Journal, December, 28, 2000).*

Information and power can also be sources of incompleteness. There are many situations when this is the case. Information asymmetries, insider trading and advantages of various sorts can provide an edge to individual investors and thereby violate the basic tenets of market efficiency. Further, the interaction of markets can lead to instabilities due to very rapid and positive feedback or to expectations that are becoming trader- and market-dependent. Such situations lead to a growth of volatility, instabilities and perhaps, in some special cases, to chaos. Nonetheless, whether it is fully right or wrong, it seems to work sometimes. Thus, it should be used carefully for making money. Of course, it is used for simple models for pricing options and derivatives in general. Throughout these approaches we shall use a known risk-free rate. In contrast, economic equilibrium theory based on the clearing of markets by equating supply to demand for all financial assets provides an equilibrium where interest rates are endogenous. It assumes however, that beliefs are homogenous, markets are frictionless (with no transaction costs, no taxes, no restriction on short sales and divisible assets) as well as competitive markets (in other words, investors are price takers) and finally it also assumes no arbitrage. Thus, general equilibrium is more elaborate than rational expectations (and arbitrage-free pricing) and provides more explicit results regarding market reactions and prices than the traditional finance-only approach [47.54].

47.2.1 Risk-Neutral Pricing and Complete Markets

In complete markets, we use risk-neutral probabilities which allow, conveniently, linear pricing measures. If there are such probabilities, and it is so in complete markets, then the current price ought to be determined by its future values. Since these probabilities are endogenous, based on an exchange between investors and speculators, *it is the market that determines prices and not uncertainty*. Uncertainty arises then only from an individual assessment of potential future events based on private information, on the one hand, or based on investors and speculators not aware of publicly available common information, on the other. In such situations the complete-markets mechanism (based on risk-neutral pricing) for determining asset prices is no longer viable.

Our ability to construct a unique set of risk-neutral probabilities depends on a number of assumptions which are of critical importance in finance and must be maintained theoretically and practically. These include: no arbitrage opportunities; no dominant trading strategies; and the law of the single price. No arbitrage occurs when it is not possible for an agent to make money for sure without having to invest any in the first place. The single-price hypothesis was elaborated by Modigliani and Miller in 1958, stating that two prospective future cash flows with identical risks must be priced equally. In other words, if we can replicate an asset price by a (synthetic) portfolio whose value can be ascertained, then if there is no arbitrage, the price of the asset and the synthetic portfolio are necessarily the same. This also implies market completeness, requiring that there be: no transaction costs; no taxes; infinitely divisible assets; that gents can borrow or lend at the same rates; no information asymmetry regarding future state prices; an impossibility to short sell; and finally, rational investors. Any violation or restrictions that will violate market completeness will open an opportunity for arbitrage. Although in practice, at least one of these assumptions is often violated, in theory and for many fundamental and useful results in finance theory, the assumption of no arbitrage is maintained. When this is not the case, and the assumption of market completeness is violated, it is no longer possible to obtain a unique set of *risk-*

neutral probabilities, but there may be several sets of such *risk-neutral* probabilities. To price an asset in such circumstances, statistical and numerical techniques are applied to select the martingale that best fits the observed behavior of financial markets and at the same time is consistent with basic economic and financial considerations. There are a number of approaches that we might follow in such circumstances, some of which will be used in the sequel. When markets are complete, replication of assets by a synthetic portfolio is a powerful tool to determine asset prices. Such as, pricing forward contracts, put–call parity, future prices and other derivatives that use replicating portfolios. For example, say that a stock spot price is S and let R_f be the period's risk-less lending rate. Next consider a forward contract consisting of an agreement to buy the stock at the end of the period at a set price F. If no dividends are paid and there is no arbitrage, then the contract profit (price) is $Q = S - F/(1+R_f) = 0$ (since the contract is costless and no money exchanges hands initially). As a result, the forward price is $F = S(1+R_f)$. Similarly, for put–call parity we consider a stock and derived call and put contracts with the same strike price and time. The former gives the right to buy the stock at the strike price and the latter the right to sell at time T and price K. If their prices at maturity are $c_T = \max(S_T - K, 0)$ and $p_T = \max(K - S_T, 0)$, respectively, then the current put price can be replicated by a portfolio consisting of the call, a risk-less bond with price K at maturity T – the option's exercise price as well as the underlying stock. This will be considered later but it suffices for the moment to state that the put (and thereby its synthetic portfolio) price at time $t = 0$ is given by $p = c - S + K\,\mathrm{e}^{-R_f T}$. Other cases will be considered subsequently. Below, we consider the implications and the mechanics of risk-neutral pricing using a number of stochastic price models.

47.2.2 Risk-Neutral Pricing in Continuous Time

The purpose of this section is to consider the mechanics of risk-neutral pricing in complete markets in a continuous time model. For simplicity, we restrict our attention to an underlying log-normal asset-price process, meaning that the rates of returns of the asset are normal with known mean and known variance. We set $S(t)$ as the asset price at time t with normally distributed rates of returns $\mathrm{d}S(t)/S(t)$. This can be written as an Ito stochastic differential equation as follows:

$$\frac{\mathrm{d}S}{S} = \alpha\,\mathrm{d}t + \sigma\,\mathrm{d}W, \quad S(0) = S_0 \quad \text{or}$$

$$S(t) = S(0)\,\mathrm{e}^{\left(\alpha - \frac{\sigma^2}{2}\right)t + \sigma W(t)}, \quad W(t) = \int_0^t \mathrm{d}W(t)\,,$$

where the rates of returns are normally distributed with mean and variance, both linear functions of time, given by αt and $\sigma^2 t$ respectively. Further, $W(t)$ denotes a Brownian motion, which is defined as normal (Wiener) prices with mean zero and variance t. Formally, we first note that with this probability measure, there is no risk-neutral pricing since:

$$\begin{aligned} S(0) &\neq \mathrm{e}^{-R_f t} E\left[S(t)\right] \\ &= \mathrm{e}^{-R_f t}\,\mathrm{e}^{\left(\alpha - \frac{\sigma^2}{2}\right)t} S(0) E\left(\mathrm{e}^{\sigma W(t)}\right) \\ &= \mathrm{e}^{(\alpha - R_f)t} S(0)\,. \end{aligned}$$

This is the case since for normal distributions $E\big(\exp[\sigma W(t)]\big) = \exp\left[(\sigma^2 t/2)\right]$. However, if we define a numeraire $W^*(t) = W(t) + \frac{\alpha - R_f}{\sigma} t$ with respect to which the risk-neutral process will be defined, then we can write the price as

$$\begin{aligned} S(t) &= S(0)\,\mathrm{e}^{\left(R_f - \frac{\sigma^2}{2}\right)t + \sigma\left[W(t) + \frac{\alpha - R_f}{\sigma} t\right]} \\ &= S(0)\,\mathrm{e}^{\left(R_f - \frac{\sigma^2}{2}\right)t + \sigma W^*(t)}\,, \end{aligned}$$

which, of course, corresponds to an underlying price process (where $\alpha = R_f$) and therefore,

$$\frac{\mathrm{d}S}{S} = R_f\,\mathrm{d}t + \sigma\,\mathrm{d}W^*(t), \quad S(0) = S_0\,,$$

and:

$$S(0) = \mathrm{e}^{-R_f t} E^*\left[S(t)\right] = \mathrm{e}^{(R_f - R_f)t} S(0) = S(0)\,,$$

where E^* is an expectation taken with respect to the (numeraire) process $W^*(t)$. Thus, the current price equals an expectation of the future price, just as the risk-neutral valuation framework indicated. It is important to remember however that the proof of such a result is based on our ability to replicate such a process by a risk-free process (and thereby value it by the risk-free rate, something that will be done later on). If such a numeraire can be defined, then of course, even an option can be valued, under the risk-neutral process, by a linear expectation. For example, for a European call option whose exercise price is K at time T, its price is necessarily:

$$C(0) = \mathrm{e}^{-R_f T} E^*\left(\max\left[S(T) - K, 0\right]\right)\,.$$

In fact, under the risk-neutral framework, any asset price equals is risk-free rate discounted expectation

under the risk-neutral distribution. For this reason, much effort is expanded, theoretically and practically, on determining the appropriate risk-neutral distribution (the martingale) that can be used to determine asset prices. In particular, it is worthwhile reconsidering the example treated by letting $\lambda = \frac{\alpha - R_f}{\sigma}$ define the market risk. In this case, the measure we have adopted above is equal to the original process plus the price of risk cumulated over time, or: $W^*(t) = W(t) + \lambda t$ and therefore: $(\alpha - R_f)\,dt + \sigma W(t) = \sigma W^*(t)$. Hence, $dS = S[\alpha\,dt + \sigma\,dW(t)] = S\sigma\,dW^*(t)$. Clearly, under the transformed measure $W^*(t)$, the stock process is a martingale but the remaining question is, can we treat this measure as a Wiener process The important theorem of Giranov allows such a claim under specific conditions, which we summarize below. Explicitly, say that Novikov's condition is satisfied, that is:

$$E\left(\exp\left(\frac{1}{2}\int_0^T \left|\frac{\alpha - R_f}{\sigma}\right| dt\right)\right) < +\infty\,.$$

Let the new measure be defined by the Radon–Nykodim derivative,

$$\frac{dP^*}{dP} = \mathbb{Z}, \quad E\mathbb{Z} = 1, \quad \text{where}$$

$$\mathbb{Z} \triangleq \exp\left[-\int_0^T \lambda\,dW(s) - \frac{1}{2}\int_0^T (\lambda)^2\,dW(s)\right],$$

$$E\mathbb{Z} = 1\,,$$

where P^* is the probability equivalent of the original measure. Note that: $P^*(A) = \int_A \mathbb{Z}(W)P(dW), \forall A \in \Im$. The Girsanov theorem then states that under these (Novikov) conditions and given the measure defined by the Radon–Nykodim derivative, the process $W^*(t)$ is a Wiener process under the measure P^*. This theorem is of course extremely important in asset pricing as it allows the determination of martingales to which risk-neutral pricing can be applied.

47.2.3 Trading in a Risk-Neutral World

Under a risk-neutral process, there is no trading strategy that can make money. To verify this hypothesis, we consider an investor's decision to sell an asset he owns (whose current price is S_0) as soon as it reaches an optimal (profit-rendering) price $S^* > S_0$. Let this profit be:

$$\pi_0 = E^*\,e^{-R_f\tau}S^* - S_0 \quad \text{with}$$
$$\tau = \text{Inf}\left\{t > 0,\, S(t) \geq S^*;\, S(0) = S_0\right\}\,,$$

where τ is the stopping (sell) time, defined by the first time that the optimal target sell price is reached. We shall prove that, under the risk-neutral framework, there is an *equivalence* between selling now or at a future date. Explicitly, we will show that $\pi_0 = 0$. Again, let the risk-neutral price process be:

$$\frac{dS}{S} = R_f\,dt + \sigma\,dW^*(t)$$

and consider the equivalent return process $y = \ln S$. By an application of Ito's calculus, we have

$$dy = \left(R_f - \frac{\sigma^2}{2}\right)dt + \sigma\,dW^*(t), \quad y(0) = \ln(S_0)$$

and

$$\tau = \text{Inf}\left\{t > 0,\, y(t) \geq \ln(S^*);\, y(0) = \ln(S_0)\right\}\,.$$

As a result, $E^*_S\left(e^{-R_f\tau}\right) = E^*_y\left(e^{-R_f\tau}\right)$, which is the Laplace transform of the sell stopping time when the underlying process has a mean rate and volatility given by $\mu = R_f - \sigma^2/2$ and σ respectively, [47.56, 58–60]:

$$g^*_{R_f}(S^*, \ln S_0) = \exp\left[\frac{\ln S_0 - \ln S^*}{\sigma^2}\left(-\mu + \sqrt{\mu^2 + 2R_f\mu\sigma^2}\right)\right],$$
$$\sigma > 0, -\infty < \ln S_0 \leq \ln S^* < \infty\,.$$

The expected profit arising from such a transaction is thus

$$\pi_0 = S^*E^*\left(e^{-R_f\tau}\right) - S_0 = S^*g^*_{R_f}(\ln S^*, \ln S_0) - S_0\,.$$

Namely, such a strategy will, in a risk-neutral world, yield a positive return if $\pi_0 > 0$. Elementary manipulations show that this is equivalent to:

$$\pi_0 > 0 \quad \text{If} \quad \frac{\sigma^2}{2} > (R_f - 1)\,\mu \quad \text{or}$$
$$\frac{\sigma^2}{2} > (1 - R_f)\left(\frac{\sigma^2}{2} - R_f\right) \quad \text{if} \quad R_f > \frac{\sigma^2}{2}\,.$$

As a result,

$$\pi_0 = \begin{cases} > 0 & \text{If } R_f > \frac{\sigma^2}{2} \\ < 0 & \text{If } R_f < \frac{\sigma^2}{2}\,. \end{cases}$$

The decision to sell now or wait is thus reduced to the simple condition stated above. An optimal selling price in these conditions can be found by optimizing the return of such a sell strategy, which is found by noting that either it is optimal to have a selling price as large as

possible (and thus never sell) or select the smallest price, implying selling now at the current (any) price. If the risk-free rate is *small* compared to the volatility, then it is optimal to wait, and vice versa, a small volatility will induce the holder of the stock to sell. In other words, for an optimal sell price:

$$\frac{\mathrm{d}\pi_0}{\mathrm{d}S^*} = \begin{cases} > 0 & R_\mathrm{f} < \sigma^2/2 \\ < 0 & R_\mathrm{f} > \sigma^2/2\,. \end{cases}$$

Combining this result with the profit condition of the trade, we note that

$$\begin{cases} \frac{\mathrm{d}\pi_0}{\mathrm{d}S^*} > 0, \pi_0 < 0 & \text{If } R_\mathrm{f} < \sigma^2/2\,, \\ \frac{\mathrm{d}\pi_0}{\mathrm{d}S^*} < 0, \pi_0 > 0 & \text{If } R_\mathrm{f} > \sigma^2/2\,. \end{cases}$$

An therefore the only solution that can justify these conditions is: $\pi_0 = 0$, implying that whether one keeps the asset or sell is irrelevant, for under risk-neutral pricing, the profit realized from trading of maintaining the stock is equivalent. Say that $R_\mathrm{f} < \sigma^2/2$ then a *wait-to-sell* transaction induces an expected trade loss and therefore it is best to obtain the current price. When $R_\mathrm{f} > \sigma^2/2$, the expected profit from the trade is positive but it is optimal to select the lowest selling price, which is of course the current price and then again, the profit transaction, $\pi_0 = 0$, will be null, as our contention states. For a risk-sensitive investor (trader or speculator), however, whose utility for money is $u(.)$, a decision to sell will be defined in terms of his preference, given by the utility function. Buy–sell strategies differ therefore because investors have preferences (utilities) that are not the same.

Example 47.2 (Buying and selling on a trinomial random walk): Consider the risk-neutral log-normal risk process:

$$\mathrm{d}S/S = R_\mathrm{f}\,\mathrm{d}t + \sigma\,\mathrm{d}W, \quad S(0) = S_0$$

and apply Ito's lemma to the transformation $y = \ln(S)$ to obtain the rate of return process:

$$\mathrm{d}y = \left(R_\mathrm{f} - \frac{1}{2}\sigma^2\right)\mathrm{d}t + \sigma\,\mathrm{d}W, \quad y(0) = y_0\,.$$

Given this normal (logarithmic) price process, consider the trinomial random-walk approximation:

$$Y_{t+1} = \begin{cases} Y_t + f_1 & \text{w. p.} \quad p \\ Y_t + f_2 & \text{w. p.} \quad 1-p-q \\ Y_t + f_3 & \text{w. p.} \quad q \end{cases}$$

Where p, q and $1-p-q$ are the probabilities that returns increase (or decrease) by, f_1, f_3, f_2 respectively. The first two moments of this process are given by

$$\begin{aligned} E\left(Y_{t+1} - Y_t\right) &= f_2 + p\left(f_1 - f_2\right) + q\left(f_3 - f_2\right) \\ &\approx \left(R_\mathrm{f} - \frac{1}{2}\sigma^2\right), \\ E\left(Y_{t+1} - Y_t\right)^2 &= f_2^2 + p\left(f_1^2 - f_2^2\right) + q\left(f_3^2 - f_2^2\right) \\ &\approx \sigma^2\,. \end{aligned}$$

Thus, an appropriate selection of the parameters p, q, f_1, f_2 and f_3 will provide an approximation to the risk-neutral pricing process. However, note that we have two known parameters (the mean and the variance of the process) while there are five parameters to choose. This corresponds to many potential discrete-time processes that can be considered as approximations to the continuous one. For this reason, a discretization of a risk-neutral process can often lead to incomplete processes (where risk-neutrality cannot be applied). In most cases, therefore, the underlying process has to be carefully applied [47.48]. For an asymmetric trinomial random walk we may set for simplicity $f_1 = 1$, $f_2 = 0$ and $f_3 = -1$, in which case

$$P\left(\Delta Y_i = +1\right) = p, \quad P\left(\Delta Y_i = -1\right) = q, \quad \text{and}$$
$$P\left(\Delta Y_i = 0\right) = r = 1 - p - q.$$

It is well known (for example see *Cox* and *Miller* [47.42], p. 75) that the probability of reaching one of the two boundaries in this case is given by,

$$P\left(Y_t = -a\right) = \begin{cases} \frac{1-(1/\lambda)^b}{1-(1/\lambda)^{a+b}} & \lambda \neq 1 \\ b/(a+b) & \lambda = 1 \end{cases},$$

$$P\left(Y_t = b\right) = \begin{cases} \frac{(1/\lambda)^b - (1/\lambda)^{a+b}}{1-(1/\lambda)^{a+b}} & \lambda \neq 1 \\ a/(a+b) & \lambda = 1 \end{cases},$$

where $\lambda = q/p$. Further, the expected first time to reach one of these two boundaries is

$$E\left(T_{-a,b}\right) = \begin{cases} \left(\frac{1}{1-r}\right)\left(\frac{\lambda+1}{\lambda-1}\right)\left(\frac{a\left(\lambda^b-1\right)+b\left(\lambda^{-a}-1\right)}{\lambda^b-\lambda^{-a}}\right) & \lambda = q/p \\ \frac{ab}{1-r} & \lambda = 1\,. \end{cases}$$

Thus, if we own an asset whose current value is null and if it is sold either when the loss incurred is $-a$ or at b when a profit is realized, then the probability of making money is $P\left(S_{T(-a,b)} = b\right)$ while the probability

of losing it is $P\left(S_{T(-a,b)}=-a\right)$, as calculated above. The expected amount of time the trade will be active is of course $E\left(T_{-a,b}\right)$. The trader profit or loss is thus a random variable given by

$$\tilde{\pi}=\begin{cases}-a & \text{w. p.} \begin{cases}\frac{1-(1/\lambda)^b}{1-(1/\lambda)^{a+b}} & \lambda\neq 1\\ b/(a+b) & \lambda=1\end{cases}\\ b & \text{w. p.} \begin{cases}\frac{(1/\lambda)^b-(1/\lambda)^{a+b}}{1-(1/\lambda)^{a+b}} & \lambda\neq 1\\ a/(a+b) & \lambda=1\,.\end{cases}\end{cases}$$

While its average return is (since the process can be considered a renewal process as well)

$$\bar{\pi}(-a,b)=\frac{bP\left(S_{T(-a,b)}=b\right)-aP\left(S_{T(-a,b)}=-a\right)}{E\left(T_{-a,b}\right)}\,.$$

In particular, when $\lambda=1$, the price process is a martingale and the long-run average profit will be null with a variance $2ab$ since:

$$E(\tilde{\pi})=\frac{(-ab+ba)}{a+b}=0\,,\quad \text{var}(\tilde{\pi})=2ab\,.$$

This also means that we cannot make money on the average with a worthless asset if the underlying price process is a (martingale) random walk (whether it is a binomial or a trinomial walk). In this circumstance, there is no free gift, an asset we receive that is worth nothing is indeed worth nothing. A risk-averse investor, will thus be better off getting rid of this asset and not sustaining the risk of losing money. For a binomial random walk, with $\lambda\neq 1$ and $r=0$ we have ([47.42], p. 31):

$$P\left(S_{T(-a,b)}=b\right)=\frac{\lambda^a-1}{\lambda^{a+b}-1}\,;$$
$$P\left(S_{T(-a,b)}=-a\right)=\frac{\lambda^{a+b}-\lambda^a}{\lambda^{a+b}-1}\,.$$

And therefore

$$E\left(T_{-a,b}\right)=\left(\frac{\lambda+1}{\lambda-1}\right)\left(\frac{a\left(\lambda^b-1\right)+b\left(\lambda^{-a}-1\right)}{\lambda^b-\lambda^{-a}}\right)\,.$$

The long-run average profit is thus

$$\bar{\pi}(-a,b)=\frac{\left[b\left(\lambda^a-1\right)-a\left(\lambda^{a+b}-\lambda^a\right)\right](\lambda-1)\left(\lambda^b-\lambda^{-a}\right)}{\left[b\left(\lambda^{-a}-1\right)+a\left(\lambda^b-1\right)\right](\lambda+1)\left(\lambda^{a+b}-1\right)}\,.$$

An optimization of the average profit over the parameters (a,b) when the underlying process is a historical process provides then an approach for selling and buying for a risk-prone trader. For a risk-neutral process ($\lambda=1$), the expected profit will be null for all values a and b.

We can extend this example by considering a trader who owns now an asset worth i_0 dollars that he intends to sell at a later date, either at a preventive loss or at a given profit level. Technically, the sell strategy consists of selling at a price $b>i_0$ for a profit of $\kappa=b-i_0$ or at a price of $a<i_0$ for a loss of $\nu=i_0-a<0$ – whichever comes first. The problems we might be concerned with are:

1. What are the optimal parameters (a,b) for an individual investor if the investor uses a risk-adjusted discount rate and at a risk-free discount rate if the underlying process is a risk-neutral process?
2. What is the risk premium of such a strategy?
3. For an averages profit criterion, what are the optimal parameters (a,b) of the trading strategy?

As stated above, the rationality of such strategies are implicit in individual investors' risk aversion, seeking to make a profit by selling at a higher price, and inversely selling at a preventive loss in case prices fall too much, generating potentially a substantial losses.

47.3 Consumption Capital Asset Price Model and Stochastic Discount Factor

Financial asset pricing is essentially based on defining an approach accounting for the time and risk preferences of future payoffs. To do so, we have sought to determine a discounting mechanism that would, appropriately, reflect the current value of uncertain payoffs to be realized at some future time. The risk-neutral asset pricing framework has provided a linear estimation rule based on the risk-free rate. Another approach considers an investor optimizing the expected utility of consumption and investment. This is also coined the consumption capital asset pricing model (CCAPM), also called the stochastic discount factor (SDF) approach [47.20]. This development will be presented through the application of Euler's equation in the calculus of variations applied

to a consumption problem resulting in a pricing formula defined by:

$$p_t = E(\tilde{M}_{t+1}\tilde{x}_{t+1}), \quad \tilde{M}_{t+1} = \frac{1}{1+\tilde{R}_{t+1}},$$

where p_t is the current asset price at time t that we seek to value, $\tilde{x}_t$ is the next-period asset returns, a random variable and $\tilde{M}_t$ is a stochastic discount factor (also called a kernel). We shall show first through a simple two-period model how a pricing formula is derived. Subsequently, we consider a general multi-period problem. It is noteworthy that in this framework the current price equals the current discounted expectation of (only) future returns.

47.3.1 A Simple Two-Period Model

The rationality of the SDF approach can be explained simply by using the following example (subsequently generalized). Say that an investor owns at a certain time t, a certain amount of money s_t, part of which is invested to buy a quantity of stock y at a price p_t, while the residual is consumed. The utility of consumption is assumed to be $u(c_t)$ where $c_t = s_t - yp_t$. A period hence, at time $t+1$, the asset price is a random variable $\tilde{x}_{t+1}$, at which time it is sold and consumed. Thus, the next-period consumption is equal to the period's current income plus the return from the investment, namely $c_{t+1} = s_{t+1} + y\tilde{x}_{t+1}$. Let the discount factor be β, expressing the subjective discount rate of the consumer. Over two periods, the investor's problem consists then of maximizing the two periods' expected utilities of consumption, given by

$$U(c_t, c_{t+1}) = u(c_t) + \beta E_t u(c_{t+1}) \quad \text{or}$$
$$U(c_t, c_{t+1}) = u(s_t - yp_t) + \beta E_t u(s_{t+1} + y\tilde{x}_{t+1}).$$

The optimal quantity to invest (i. e. the number of shares to buy), found by maximizing the expected utility with respect to y, leads to:

$$\begin{aligned}\frac{\partial U}{\partial y} &= -p_t u'(s_t - p_t y) \\ &\quad + \beta E_t\left[\tilde{x}_{t+1}u'(s_{t+1} + \tilde{x}_{t+1}y)\right] \\ &= -p_t u'(c_t) + \beta E_t\left[\tilde{x}_{t+1}u'(c_{t+1})\right] = 0\end{aligned}$$

which yields for an optimum portfolio price:

$$p_t = E_t\left(\beta\frac{u'(c_{t+1})}{u'(c_t)}\tilde{x}_{t+1}\right).$$

If we set $\tilde{M}_{t+1} = \beta\frac{u'(c_{t+1})}{u'(c_t)}$, we obtain the pricing kernel (stochastic discount factor) used above to price the asset. This kernel expresses, as seen in our condition for optimality, the inter-temporal substitution of current and future marginal utilities of consumption. If we choose to write this term as a *discount rate*, then:

$$\tilde{R}_{t+1} = \left(1 - \beta\frac{u'(c_{t+1})}{u'(c_t)}\right) \Big/ \beta\frac{u'(c_{t+1})}{u'(c_t)}.$$

This equation is particularly robust, and has many well-known results in finance expressed as special cases [47.20]. For example, if the utility function is of the logarithmic type, $u(c) = \ln(c)$ then, $u'(c) = 1/c$ and $\tilde{M}_{t+1} = \beta c_t/(c_{t+1})$, or $\tilde{R}_{t+1} = \beta[(1/\beta)c_{t+1} - c_t]/c_t$ and further, $p_t/c_t = \beta E_t\left(\tilde{x}_{t+1}/c_{t+1}\right)$. In other words, if we write $\pi_t = p_t/c_t$; $\tilde{\pi}_{t+1} = \tilde{x}_{t+1}/c_{t+1}$, then we have: $\pi_t = \beta E_t\left(\tilde{\pi}_{t+1}\right)$. Further, if the asset is a risk-less bond, worth 1 dollar at its exercise time a period hence, then $B_0 = E(\tilde{M}_1 B_1)$ or $B_0 = E(\tilde{M}_1)$. Since $B_0 = 1/(1+R_\mathrm{f})$, we obtain of course: $E(\tilde{M}_1) = 1/(1+R_\mathrm{f})$, providing thereby a relationship between the expected value of the kernel and the risk-free discount rate. An additional and particularly interesting case consists of using the linear CAPM model for pricing risky assets. In this case, let the kernel be

$$M_{t+1} = a_t + b_t R_{M,t+1},$$

where $R_{M,t}$ is the rate of return of the market portfolio (a market index for example). For a given stock, whose rate of return is $1 + R_{t+1} = p_{t+1}/p_t$, we have as stated earlier:

$$\begin{aligned}&1 = E\left[M_{t+1}(1+R_{t+1})\right], \quad \text{hence} \\ &E(1+R_{t+1}) = \frac{1}{E(M_{t+1})} - \frac{\mathrm{cov}\,(M_{t+1}, 1+R_{t+1})}{E(M_{t+1})}.\end{aligned}$$

Inserting the linear model for the kernel we have

$$\begin{aligned}&E(1+R_{t+1}) \\ &= (1+R_{\mathrm{f},t})\left[1 - \mathrm{cov}\,(M_{t+1}, 1+R_{t+1})\right] \\ &= (1+R_{\mathrm{f},t+1})\left[1 - \mathrm{cov}\left(a + bR_{M,t+1}, 1+R_{t+1}\right)\right]\end{aligned}$$

which is reduced to

$$\begin{aligned}&E(R_{t+1} - R_{\mathrm{f},t+1}) \\ &= \frac{\mathrm{cov}\left(R_{M,t+1} - R_{\mathrm{f},t+1}, R_{t+1} - R_{\mathrm{f},t+1}\right)}{\mathrm{var}\left(R_{M,t+1} - R_{\mathrm{f},t+1}\right)} \\ &\quad \times E_t\left(R_{M,t+1} - R_{\mathrm{f},t+1}\right).\end{aligned}$$

This can be written in the CAPM standard formulation (see also [47.61–63]):

$$E(R_{t+1} - R_{f,t+1}) = \beta E_t\left(R_{M,t+1} - R_{f,t+1}\right).$$

where the beta parameter is:

$$\beta = \frac{\operatorname{cov}\left(R_{M,t+1} - R_{f,t+1},\, R_{t+1} - R_{f,t+1}\right)}{\operatorname{var}\left(R_{M,t+1} - R_{f,t+1}\right)} .$$

Of course when the returns are normally distributed such calculations are straightforward and can be generalized further. *Stein* [47.64] has shown that, if market returns are some function $f(y)$, $y = R_{M,t+1} - R_{f,t+1}$ and, if the derivative $f'(.)$ exists, then

$$\operatorname{cov}\left[x,\, f(y)\right] = E\left[f'(y)\right] \operatorname{cov}(x,\, y) .$$

And as a result, the beta parameter is

$$\begin{aligned}\beta &= \frac{\operatorname{cov}\left[f(R_{M+1} - R_{f,t+1}),\, R_{t+1} - R_{f,t+1}\right]}{\operatorname{var}\left(R_{M+1} - R_{f,t+1}\right)} \\ &= \frac{E\left[f'(R_{M+1} - R_{f,t+1})\right]}{\operatorname{var}\left(R_{M+1} - R_{f,t+1}\right)} \\ &\quad \times \operatorname{cov}\left[f'(R_{M+1} - R_{f,t+1}),\, R_{t+1} - R_{f,t+1}\right] .\end{aligned}$$

For example, in a stochastic inflation world, *Roll* [47.65] extends the CAPM of Sharpe by setting:

$$E(RP) = R_{\mathrm{f}} E(P) + \beta E\left(R_M P - R_{\mathrm{f}} P\right) ,$$

where P is a stochastic purchasing power with

$$\beta = \frac{\operatorname{cov}\left(RP,\, R_M P\right)}{\operatorname{var}\left(R_M P\right)} .$$

The hypothesis that the kernel is linear may be limiting however. Recent studies have suggested that we use a quadratic measurement of risk with a kernel given by:

$$M_{t+1} = a_t + b_t R_{M,t+1} + c_t R_{M,t+1}^2 .$$

In this case, the skewness of the distribution enters as well in determining the value of the asset. There is ongoing empirical research on this and related topics.

47.3.2 Euler's Equation and the SDF

The CCAPM model, in its inter-temporal framework can be formulated as a problem in the calculus of variations and the SDF determined by applying the Euler condition for optimal consumption utility [47.1]. Explicitly, let an investor maximize the expected utility of consumption over a horizon $[0, T]$:

$$V_t = \max \sum_{j=0}^{T-1} \beta^j u(c_{t+j}) + \beta^T G(S_T) ,$$

where $u(c_{t+j})$ is the utility of consumption c_{t+j} at time $t+j$, T is the final time and $G(S_T)$ expresses the terminal utility of the wealth state at time T. The investor's discount rate is β. At time t, the investor's wealth is given by $S_t = S_{t-1} - q_t c_t + R_t$, where consumption is priced q_t, while returns from investments are R_t. As a result:

$$c_{t+j} = \frac{R_{t+j} - \Delta S_{t+j}}{q_{t+j}}, \quad \Delta S_{t+j} = S_{t+j} - S_{t+j-1} .$$

The investor's utility is therefore:

$$V_t = \max \sum_{j=0}^{T-1} \beta^j u\left(\frac{R_{t+j} - \Delta S_{t+j}}{q_{t+j}}\right) + \beta^T G(S_T) .$$

Application of Euler's equation (the calculus of variations) yields:

$$\frac{\partial V_t}{\partial S_{t+j}} - \Delta\left(\frac{\partial V_t}{\partial \Delta S_{t+j}}\right) = 0 .$$

Since $\partial V_t / \partial S_{t+j} = 0$, $\Delta\left(\partial V_t / \partial \Delta S_{t+j}\right) = 0$ and therefore we have the following *equilibrium* results:

$$\frac{\partial V_t}{\partial \Delta S_{t+j}} = \frac{\beta^j}{q_{t+j}} \frac{\partial u\left(c_{t+j}\right)}{\partial \Delta S_{t+j}} = A \text{ constant} .$$

For two consecutive instants of time ($j = 0$, $j = 1$):

$$\begin{aligned}\frac{\partial V_t}{\partial \Delta S_t} &= \frac{\partial V_t}{\partial \Delta S_{t+1}} \quad \text{and therefore} \\ \frac{1}{q_t} \frac{\partial u\left(c_t\right)}{\partial \Delta S_t} &= \frac{\beta}{q_{t+1}} \frac{\partial u\left(c_{t+1}\right)}{\partial \Delta S_{t+1}} \quad \text{and} \\ \frac{\partial u\left(c_t\right)}{\partial \Delta S_t} &= \beta E\left(\frac{q_t}{q_{t+1}} \frac{\partial u\left(c_{t+1}\right)}{\partial \Delta S_{t+1}}\right) .\end{aligned}$$

In other words, the marginal utility of wealth (savings) equals the discounted inflation-adjusted marginal utilities of consumption. In particular, if wealth is invested in a portfolio of assets such that:

$$\begin{aligned}\Delta S_t &= (N_t - N_{t-1}) p_t = p_t \Delta N_t \quad \text{and} \\ \frac{\partial V_t}{\partial \Delta S_{t+1}} &= p_t \frac{\partial u\left(c_{t+1}\right)}{\partial \Delta N_{t+1}} \quad \text{then} \\ \frac{\partial u\left(c_t\right)}{\partial \Delta N_t} p_{t-1} &= E\left[\beta \frac{q_t}{q_{t+1}} \left(\frac{\partial u\left(c_{t+1}\right)}{\partial \Delta N_{t+1}} p_t\right)\right] ,\end{aligned}$$

which is reduced to the previous condition in two periods, or

$$\begin{aligned}p_{t-1} &= E\left(\beta \frac{q_t}{q_{t+1}} \frac{u'\left(c_{t+1}\right)}{u'\left(c_t\right)} p_t\right) ; \\ u'\left(c_{t+1}\right) &= \frac{\partial u\left(c_{t+1}\right)}{\partial \Delta N_{t+1}} .\end{aligned}$$

We can write this expression in terms of the stochastic factor M_t, expressing again the *consumption impatience*

$$p_{t-1} = E\,(M_t p_t)\,; \quad M_t = \beta \frac{q_t}{q_{t+1}} \frac{u'\,(c_{t+1})}{u'\,(c_t)}\,.$$

Again, if we set $1 + R_t = p_t / p_{t-1}$, this equation can also be written as follows:

$$1 = E\left(M_t \frac{p_t}{p_{t-1}}\right), \quad \text{hence}$$
$$1 = E\,\{M_t(1+R_t)\,|\Phi_t\}\,,$$

which is the standard form of the SDF equation while Φ_t is a filtration at time t. Here too, we see that to price a default-free zero-coupon bond paying one dollar for sure at time 1, then applying the known risk-free discount rate R_f, we have

$$\frac{1}{1+R_\mathrm{f}} = (1)E\,(M_t)\,. \quad \text{And therefore,}$$
$$E\,(M_t) = \frac{1}{1+R_\mathrm{f}} \quad \text{and finally}$$
$$p_{t-1} = \frac{1}{1+R_\mathrm{f}} E_t^*\,(p_t)\,,$$

where E_t^* assumes the role of a modified (subjective) probability distribution. When the utility function is assumed known, some simplifications can be reached. For example, for $u(.) = \ln(.)$, $u'\big(c_t\big) = 1/c_t$ and therefore $v_t = E\big\{\beta(q_t/q_{t+1})v_{t+1}\big\}$, $v_t = p_{t-1}/c_t$.

47.4 Bonds and Fixed-Income Pricing

The financial valuation of assets, real or financial, deals with streams of cash such as dividends, coupon payments, investment in engineering projects etc. which occur in a random manner or not, paid in at deterministic or random times. In some cases, there may be a default in such payments due to delays, lost and partially recuperated payments etc. For example, investing in a portfolio might result in future returns and dividends that are at best defined in terms of random cash flows. Traditionally, a number of techniques were applied to value such cash streams, spanning a broad set of subjective techniques such as: payback, internal rate of return (IRR), cost–benefit analysis (CBA), net present value (NPV) etc. Bonds pricing is often used to value these cash flows. Here we shall see how bonds, whether risk-free, rated or default-prone, are used to value these cash flows. The simplest bond is the zero-coupon risk-free bond paying 1 dollar a year from now. An investor can have an individual valuation of such a payment, say $B_\mathrm{IND} = 1(1+r)^{-1}$, in which case r represents the discount factor that the investor is willing to associate to such a payment. Buying such a bond is an investment in a risk-free payment which cannot earn anything else but the risk-free rate (otherwise there would be arbitrage). Say that the *market price* for such a bond is currently quoted at \$ 0.90. In this case, the discount rate that the market associates to this bond would be:

$$B_\mathrm{IND} = 0.90 = \frac{100}{(1+R_\mathrm{f})} \quad \text{or}$$
$$R_\mathrm{f} = \frac{1.0}{0.90} - 1 = 0.1111\,,$$

where R_f is used to denote the fact that this is a risk-free rate (since the bond payment has no risk). These rates are usually specified by US government bonds when they are assumed to be risk-less (a currency trader might not think this is the case, however). The price of this bond is usually specified by a function $B_\mathrm{f}(t, T)$, which is the price at time t for a bond paying 1 dollar for sure at time T when the going risk-free rate $R_\mathrm{f}(t, T)$ expresses the time structure of interest rates. For a stream of payments, say a project defined by a set of payments and returns in the future, corporate firms may use a discount rate r for the time value of money. In this case, the present value of such a project at the initial time $t = 0$, NPV(0), is written as follows:

$$\mathrm{NPV}(0) = -\sum_{i=0}^{n} \frac{I_i}{(1+r)^i} + \sum_{i=0}^{N} \frac{C_i}{(1+r)^i}\,,$$

where I_i denotes the investment (or costs sustained at time i, while C_i is a certain (risk-free) cash flow generated by the project. If $\mathrm{NPV}(0) = 0$ then the solution of this equation yields the IRR, the corporate entity uses to rank and value investment projects. There are many problems with this valuation however that open an opportunity for arbitrage by investment funds. For example, the investments and return (or the Is and Cs) may be random, potentially involving defaults, payments delays and so on. Further, the discount rate used might not reflect the cost of borrowing of the firm and its risk rating (potentially given by rating firms such as Moody's, Fitch, Standard and Poor, and their like). In addition, the discount rate does not reflect the time at

which these payments occur (the term structure). As a result, such a valuation (pricing) is quite naïve and arbitrage on these firms can be used to provide the same cash flows at a lower price, thereby cashing in the difference. If payments are known for sure, a market-sensitive valuation would use the term structure risk-free discount rate $R_f(0, i)$ for the payment i-periods hence, the value of such a cash stream would be:

$$\mathrm{NPV}_f(0) = -\sum_{i=0}^{n} \frac{I_i}{[1 + R_f(0, i)]^i} + \sum_{i=0}^{n} \frac{C_i}{[1 + R_f(0, i)]^i} .$$

While at any time t, it is given by:

$$\mathrm{NPV}_f(t) = -\sum_{i=t}^{n} \frac{I_i}{[1 + R_f(t, i)]^{i-t}} + \sum_{i=t}^{n} \frac{C_i}{[1 + R_f(t, i)]^{i-t}} .$$

This latter expression can of course be written in terms of zero-coupon risk-free bonds as follows:

$$\mathrm{NPV}_f(t) = -\sum_{i=t}^{n} I_i B_f(t, i) + \sum_{i=t}^{n} C_i B_f(t, i) .$$

If all pure bonds are priced by the market then of course the NPV is determined by the market. In practice however, pure bond prices are available for only a given subset of times and therefore the NPV has to be priced in some other manner. Subsequently, we shall see that this leads to an important technical problem in financial engineering – one of calculating the yield of the bond (or any portfolio).

Next, say that payments are made by a firm rated k, in which case, the project NPV can be written as follows:

$$\mathrm{NPV}_k(0) = -\sum_{i=0}^{n} \frac{I_i}{[1 + R_f(0, i)]^i} + \sum_{i=0}^{N} \frac{\hat{C}_i}{[1 + R_k(0, i)]^i}, \quad \hat{C}_i = E\left(\tilde{C}_i\right),$$

where $R_k(0, i)$ is the discount rate applied for expected receipts $\hat{C}_i$, i periods hence for a firm whose risk notation is k (for example, AAA, BB, B+, C etc.). Note that in the above expression, we have maintained the payments I_i deterministic and therefore they ought to be discounted at the risk-free rate in effect at time $t = 0$ for time i (if this is not the case, then of course, it will be necessary to select the appropriate discount rate as well). Note that $R_k(0, i)$ expresses the k-rated firm's term structure used for discounting its future returns which includes the risk premium $\Delta_k(0, i)$ in the firm's cash flows, or $R_k(0, i) = R_f(0, i) + \Delta_k(0, i)$. For example, say that the firm is currently rated k. The implication of such a rating is that an obligation of the firm to pay in i periods 1 dollar is currently priced by the market at $B_k(0, i)$. As a result, the set of future prospective investment returns of the firm can be priced by:

$$\mathrm{NPV}_k(0, n) = -\sum_{i=0}^{n} I_i B_f(0, i) + \sum_{i=0}^{n} \hat{C}_i B_k(0, i) ,$$

and at time t,

$$\mathrm{NPV}_k(t, n) = -\sum_{i=t}^{n} I_i B_f(t, i) + \sum_{i=t}^{n} \hat{C}_i B_k(t, i) .$$

This NPV includes of course the discount rate that the market applies to the firm's obligations. For fixed and secured payments the firm is obliged to use the risk-free rate, or equivalently it equals a risk-free coupon bond paying one dollar i-periods hence and denoted by $B_f(0, i)$. When we use the same discount rate for certain payments and uncertain (valued at expectation) costs and returns, the traditional approach may overestimate (or underestimate) the net present value of the investment. For example, a firm which is highly rated may be tempted to borrow more money because it is cheaper than say another firm perceived as risky. By the same token, investment in some projects (ports, highways etc.) may be less expensive when they are performed by a government, that can tax its citizen to repay a loan taken to build such a project, than say, a firm who would invest to self-build the project. Of course, it is for these reasons that private investors in national projects require some government assurance and insurance to reduce the risk (and thereby the risk premium) which they have to pay for building such projects.

The approach outlined above can be generalized further and applied to value all kinds of assets; we consider some examples. Again let $\mathrm{NPV}_k(t)$ be the net present value of an investment project at time t when the firm is rated k. Such a firm may however switch to being rated j with probability p_{kj} a period of time (year) later. These probabilities define a Markov chain $\boldsymbol{P} = [p_{kj}]$, usually specified by rating firms (Moody's, Standard and Poor, Fitch etc.). As a result, over two consecutive periods, we

have

$$\mathrm{NPV}_k(t) = -I_t + C_t + B_k(t, t+1) \times \sum_{\ell=1}^{m} p_{k\ell}\mathrm{NPV}_\ell(t+1)$$

and at the final time n, the NPV is given as a function of the rating (risk) state the firm will be in and specified by $\mathrm{NPV}_\ell(n)$. Further, note that due to the potential (or non-)transition of the firm's rating, the price of the bond may be altered over time as well. Explicitly, the price $B_k(t, T)$ of a coupon paying bond $c_{k,t}$ at time t and l_k dollars at maturity when it is rated k equals the expected present value of the bond in the next period, discounted at a rate associated with its rating in the next period. This is given by the following recurrence equation

$$B_k(t, T) = c_{k,t} + \sum_{j=1}^{m} \frac{p_{kj}}{1+R_{jt}} B_j(t+1, T)\,;$$
$$B_k(T, T) = 1_k, \quad k = 1, 2, 3, \ldots m\,.$$

However, note that for the NPV valuation, we used a zero-coupon bond paying one dollar in the next period, and therefore,

$$B_k(t, t+1) = \sum_{j=1}^{m} \frac{p_{kj}}{1+R_{jt}} B_j(t+1, t+1) = \sum_{j=1}^{m} \frac{p_{kj}}{1+R_{jt}} 1_j\,.$$

For example, if the bond pays one dollar in all circumstances, except if it is rated m, in which case it pays nothing, then

$$B_k(t, t+1) = \sum_{j=1}^{m-1} \frac{p_{kj}}{1+R_{jt}} 1 = \sum_{j=1}^{m-1} \frac{p_{kj}}{1+R_{jt}}\,.$$

Now assume that at some future time t we have the option to stop the bond payments at a price of say $-Q_t$. In other words, the actual net present value at time t with an option to stop at this time would be:

$$\mathrm{NPV}_k^{(o)}(t) = \max\left[-Q_t, \mathrm{NPV}_k(t)\right]\,,$$
$$\mathrm{NPV}_k^{(o)}(T) = \mathrm{NPV}_k(T)$$

and Q_t is the cost associated with implementing the option (for example, the cost to the firm of discontinuing a service, etc.). Note that at the final time, the option is worthless if the project has been terminated. A stopping time (at which the option is exercised) occurs thus at time τ when $-Q_t \geq \mathrm{NPV}_k(\tau)$. A number of situations may arise then. For example, for a firm that is down-rated, the cost of borrowing would increase and it might lead it to decide to exercise the option because of its cost in capital. And vice versa, a firm that is up-rated and is trapped in a costly investment might decide to either stop it or refinance it to reduce its cost or have the current cash flow of the project to be more in concordance with its improved rating. The price of such a *real* option can be valued by noting that, if it is exercised at time $t+1$, when the firm is rated j, resulting in a savings of $-Q_{t+1} - \mathrm{NPV}_j(t+1)$, with $(\mathrm{NPV}_j < 0)$, this saving is worth today

$$\mathrm{NPV}_k^{(1)}(t) = B_k(t, t+1) \sum_{\ell=1}^{m} p_{k\ell} \times \max\left(-Q_{t+1} - \mathrm{NPV}_j(t+1), 0\right)$$
$$\mathrm{NPV}_k^{(1)}(\tau) = -Q_\tau - \mathrm{NPV}_j(\tau)\,, \quad \tau \leq T\,.$$

These equations are of course meaningful only when the discount rate associated to a given rating is specified. If this is the case, then of course, our equations are simple to calculate. A potential for arbitrage may thus occur when these discount rates are not appropriately specified. Later on, we shall be concerned with the determination of these rates based on the construction of a bonds portfolio of various ratings. Examples that demonstrate how calculations are performed will also be used.

Finally, it is worth noting that, when a stream of payments are random, given by $\tilde{C}_i$, and subjectively valued by an investor whose utility of money $u(.)$ is known, then we can calculate the certainty equivalent CE_i of the uncertain payment, in which case: $u\left(CE_i\right) = Eu\left(\tilde{C}_i\right)$ and $CE_i = u^{-1}\left[Eu\left(\tilde{C}_i\right)\right]$. The NPV can then be calculated by applying the risk-free rate:

$$\mathrm{NPV}_{\mathrm{f}}(0) = -\sum_{i=0}^{n} \frac{I_i}{[1+R_{\mathrm{f}}(0,i)]^i} + \sum_{i=0}^{n} \frac{u^{-1}\left[Eu\left(\tilde{C}_i\right)\right]}{[1+R_{\mathrm{f}}(0,i)]^i}\,.$$

Unfortunately, this valuation is also subjective for it is based on a utility function which might not be available. Alternatively, a market valuation, can be used when the price of risk is known, or we establish some mechanism for appropriately accounting for the risk implied in an uncertain cash flow. This is done by calculating the yield of a bond. There are numerous techniques, inspired both

theoretically and numerically, that are applied to calculate the yield. Such a problem is the topic of commercial and theoretical research. Nonetheless we shall consider a number of approaches to calculating the yield because of its importance in financial engineering.

47.4.1 Calculating the Yield of a Bond

The yield of a bond is the discount rate y_T applied to holding the bond for T periods. This yield is often difficult to calculate because data pertaining to the term structure of zero-coupon bonds is simply not available, or available only for certain periods. For example, say that we have a bond at time whose exercise price occurs at time t, or $B(0,t) = B(t)$. To each time t, we associate the rate $y(t)$ and therefore the bond price is $B(t) = \exp\left[-y(t)t\right]$. The function $y(t)$ is called the yield curve. Of course, if zero-coupon bonds are price for time $t = 1$, we then have $y(1) = R_f$, which is the current spot rate. However, if there are no zero-coupon bonds for $t = 6$, the yield for such a bond can only be inferred by some numerical or estimation technique. In other words, unless a zero-coupon bond exists for every maturity for which the discount factor is desired, some estimation technique will be needed to produce a discount factor for any *off-maturity* time. Since zero-coupon bonds are available for only some and other maturities, a lack of liquidity may prevent the determination of the true bond yield [47.66] (see also [47.67–69]. The *yield engineering* problem consists then in determining some technique and appropriate sources of information to estimate the yield for all maturities (also called the yield curve). Approaches to this problems are of course varied. *Nelson* and *Siegel* [47.70] for example suggest the following four-parameter equation, which can be estimated numerically by fitting to the appropriate data

$$y(t) = a_0 + (a_1 + a_2)\left(\frac{1-\exp(-a_3 t)}{a_3}\right) - a_2 \exp(-a_3 t) \; ;$$

$y(t)$ is the spot rate while a_0, a_1, a_2, a_3 are the model's parameters (for related studies and alternative models see [47.66, 71–79], www.episolutions.com) suggest however a zero curve solution, which uses a combination of liquid securities, both zero-coupon and coupon-bearing bonds for which prices are readily available, and consisting of an application of bootstrapping techniques to calculate the yield curve. Explicitly, the Wets approach is based on an approximation, and in this sense it shares properties with purely spline methods. It is based upon a Taylor series approximation of the discount function in integral form. Some prevalent methods for computing (extracting) the zeros, curve-fitting procedures, and equating the yield curve to observed data in central banks include, among others: in Canada the use of the Svensson procedure and David Bolder (Bank of Canada), in Finland the Nelson–Siegel procedure, in France the Nelson–Siegel, Svensson procedures, while in Japan and the USA, the banks use smoothing splines etc. (see [47.71–73, 80]). A critical appreciation of the zero-curve approach is provided by [47.81] and [47.80] (essentially based on the structural form of the polynomial used in the episolutions approach). Other approaches span numerical techniques, smoothing techniques, [47.82], (such as least-squares approaches as we shall see later on when introducing rated bonds), kernel smoothing (SDF) techniques etc. [47.83–86]. For example, consider the price $B(t, T-t)$ of a bond at time t whose maturity is at time T. The next-period price of the bond is in fact unknown (depending on numerous factors including random interest rates). Applying the SDF approach, we can state that

$$B(t, T-t) = E_t\left[M_{t+1} B(t+1, T-t-1)\right] ,$$

where M_{t+1} is the pricing kernel. Rearrange this term as follows

$$1 = E_t\left(M_{t+1}\frac{B(t+1, T-t-1)}{B(t, T-t)}\right) , \quad \text{where}$$
$$\frac{B(t+1, T-t-1)}{B(t, T-t)} = 1 + y_{t+1,T}$$

with $y_{t+1,T}$ denoting the yield of the bond whose maturity is at time T, at time $t+1$. As a result,

$$1 = E_t\left[M_{t+1}\left(1 + y_{t+1,T}\right)\right] ,$$
$$E_t\left[M_{t+1}\right] = \frac{1}{1+R_{t,f}} .$$

To calculate the yield some model is needed for both the kernel and of course the yield distributions. Since these variables are both random and dependent an appropriate model has to be constructed on the basis of which an empirical econometric verification can be reached. Alternatively, if information regarding the marginal distributions of the kernel and the returns is available, then we may also construct copulas to represent the statistical covariation effects of both the kernel and the returns. These are problems that require further research however.

47.4.2 Bonds and Risk-Neutral Pricing in Continuous Time

In many situations, we use stochastic models of interest rates to value bonds. Below we shall consider some examples and provide as well some general results for the valuation of such bonds. In practice however, it is extremely difficult to ensure that such models do indeed predict very well the evolution of interest rates and therefore there is a broad range of techniques for calculating the *yields* of various bonds – of both risk denomination and term structures. In continuous time, let $r(t)$ be the known spot interest rate. A risk-free bond paying one dollar at T with a compounded interest rate $r(t)$ is then given by:

$$B(0,t) = \exp\left[\int_0^t r(u)\,\mathrm{d}u\right], \quad B(T,T) = 1\,.$$

The interest rate process may be deterministic or stochastic as stated above. Since bonds depend intimately on the interest rate process, it is not surprising that much effort is devoted to constructing models that can replicate and predict reliably the evolution of interest rates, as one process values the other. There are many interest rate models however, each expressing an economic rationale for the evolution of interest rates. Generally, and mostly for convenience, an interest rates process $\{r(t), t \geq 0\}$ is represented by an Ito stochastic differential equation:

$$\mathrm{d}r = \mu(r,t)\,\mathrm{d}t + \sigma(r,t)\,\mathrm{d}w\,,$$

where μ and σ are the drift and the diffusion function of the process which may or may not be stationary. Various authors consider alternative models in their analysis [47.1, 87, 88]. The Vasicek model in particular provides a straightforward rationality for interest rates movements (also called the Ornstein–Uhlenbeck process). In other words, it states that the rate of change in interest rates fluctuates around a long-run rate α. This fluctuation is subjected to random and normal perturbations of mean zero and variance $\sigma\Delta t$, or

$$\mathrm{d}r = \beta(\alpha - r)\,\mathrm{d}t + \sigma\,\mathrm{d}w\,.$$

Without much difficulty, it can be shown (see also [47.48]) that this equation has a solution expressed in terms of the current interest rates and the model's parameters given by

$$r(\tau) = \alpha + [r(t) - \alpha]\,\mathrm{e}^{-\beta(\tau - t)} + \sigma \int_t^\tau \mathrm{e}^{-\beta[\tau - y]}\,\mathrm{d}w(y)\,.$$

The value of a bond with variable interest rates is thus:

$$\begin{aligned} B(t,\tau) &= E\exp\left[\int_t^\tau r(u)\,\mathrm{d}u\right] \\ &= E\exp\left\{\int_t^\tau \alpha + [r(t) - \alpha]\,\mathrm{e}^{-\beta(u-t)} + \sigma\int_t^u \mathrm{e}^{-\beta[u-y]}\,\mathrm{d}w(y)\,\mathrm{d}u\right\}; \\ B(T,T) &= 1 \end{aligned}$$

with $\mathrm{d}w(y)$ denoting the risk source (a normally distributed random variable of zero mean and variance $\mathrm{d}y$). Interest rates are therefore also normal with a mean and variance (volatility) evolution we can easily compute. In particular note that:

$$\ln B(t,\tau) = \ln\left\{\int_t^\tau \alpha + [r(t) - \alpha]\,\mathrm{e}^{-\beta(u-t)}\,\mathrm{d}u\right\} + \ln E\exp\left[\int_t^\tau \sigma \int_t^u \mathrm{e}^{-\beta[u-y]}\,\mathrm{d}w(y)\,\mathrm{d}u\right]$$

which can be written as a linear function in the current interest rate, or

$$\ln B(t,\tau) = A(t,\tau)r(t) + D(t,\tau)\,.$$

This is a general property called the affine property which, is found in some general Markov processes X in a state space $D \subset \mathbb{R}^d$. Namely, it states that the bond return is linear in the process X, or $R(x) = a_0 + a_1 X$. Explicitly, we have the characteristic function:

$$E\left\{\mathrm{e}^{\mathrm{i}uX(t)}|X(s)\right\} = \exp[(\varphi t - s, u) + \psi(t-s,u)X(s)]\,.$$

The logarithm is of course a linear function with $a_0 = \varphi(t-s,u)$ and $a_1 = \psi(t-s,u)$ deterministic coefficients. *Duffie* et al. [47.89] show that for a time-homogeneous affine process X with a state space of the form $\mathbb{R}_+^n \times \mathbb{R}^{d-n}$, provided the coefficients $\varphi(\cdot)$ and $\psi(\cdot)$

of the characteristic function are differentiable and their derivatives are continuous at 0. The affine process X must a be a jump-diffusion process in that

$$\mathrm{d}X_t = \mu(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t + \mathrm{d}J_t$$

for standard Brownian motion W in $\mathbb{R}^d$ and a pure jump process J, with J affine dependent on X. A related property is of course

$$E_t\left(\mathrm{e}^{\int_t^s -R[x(u)]\,\mathrm{d}u + wX(s)}\right) = \mathrm{e}^{\alpha(s-t)+\beta(s-t)X(t)}\,,$$

where $\alpha(\cdot)$, $\beta(\cdot)$ satisfy a generalized Ricatti ordinary differential equation (with real boundary conditions). To see this property (in a specific case) consider the following example. Let interest rates be given by the following stochastic differential equation

$$\mathrm{d}r = \beta(\alpha - r)\,\mathrm{d}t + \sigma\sqrt{r}\,\mathrm{d}w\,.$$

Application of Ito's differential rule to $B(t,r) = \exp\left[-\int_t^T r(u)\,\mathrm{d}u\right]$ yields

$$\frac{E\,\mathrm{d}\ln B(t,r)}{\mathrm{d}t} = -r - (T-t)\beta(\alpha - r) + \frac{1}{2}(T-t)^2\sigma^2 r\,,$$

which is clearly a linear function of the current interest rate. Elementary mathematical treatment will also show that the mean and the variance of the interest rates are given by

$$E\{r(t)\mid r_0\} = c(t)\left(\frac{4\beta\alpha}{\sigma^2} + \xi\right)\,;$$

$$\mathrm{var}\,\{r(t)\mid r_0\} = c(t)^2\left(\frac{8\beta\alpha}{\sigma^2} + 4\xi\right), \quad \text{where}$$

$$c(t) = \frac{\sigma^2}{4\beta}\left(1 - \mathrm{e}^{-\beta t}\right)\,,$$

$$\xi = \frac{4r_0\beta}{\sigma^2[\exp(\beta t) - 1]}\,.$$

In this case, interest rates are not normal. Nonetheless the Laplace transform can be calculated and applied to price the bond as we have shown it above.

When interest rate models include stochastic volatility, the valuation of bonds is incomplete. Therefore, it is necessary to turn to appropriate mechanisms that can help us to price bonds. For example, denote by $V = \sigma^2(r,t)$, a *stochastic volatility model* consisting of two stochastic differential equations, with two sources of risk (W_1, W_2), which may be correlated or not. An example would be

$$\mathrm{d}r = \mu(r,t)\,\mathrm{d}t + \sqrt{V(r,t)}\,\mathrm{d}W_1\,;$$
$$\mathrm{d}V = \nu(V,r,t)\,\mathrm{d}t + \gamma(V,r)\,\mathrm{d}W_2\,,$$

where the variance V appears in both equations. Due to market incompleteness, there may be an infinite number of prices. A special case provided by *Hull* and *White* [47.90] is reproduced below. Note that the interest rate model is the square-root model we saw earlier. However, since the variance is subject to stochastic variations as well, it is modeled separately as a stochastic differential equation which is *mean–variance* reverting.

$$\frac{\mathrm{d}r}{r} = \mu\,\mathrm{d}t + \sqrt{V}\,\mathrm{d}W_1\,;$$
$$\mathrm{d}V = \alpha(\beta - V)\,\mathrm{d}t + \gamma r V^{\lambda}\,\mathrm{d}W_2\,.$$

In this case, when stock prices increase, volatility increases, while when volatility increases, interest rates (or the underlying asset we are modeling) increases also. These problems will be considered subsequently when we treat incomplete markets.

47.4.3 Term Structure and Interest Rates

If $r(t,T)$ is the interest rate applied at t for a payment at time T, then at $t+1$, the relevant rate for this period T would be $r(t+1,T)$. If these interest rates are not equal, there may be an opportunity for refinancing [47.91]. As a result, the evolution of interest rates for different maturity dates is important. Further, since bonds may have various maturities, the interest rates applied to value these bonds require necessarily that we assess the interest rates term structure. Below, we shall see how the term structure is implicit in bonds valuation. Say that an interest rate model for maturity at T is:

$$\mathrm{d}r(t,T) = \mu(r,T)\,\mathrm{d}t + \sigma(r,T)\,\mathrm{d}W\,.$$

A bond price with the same maturity is therefore a function of such interest rates, leading to:

$$\frac{\mathrm{d}B(t,T)}{B(t,T)} = \alpha(r,t,T)\,\mathrm{d}t + \beta(r,t,T)\,\mathrm{d}W\,.$$

The parameters $\alpha(.)$ and $\beta(.)$ are easily found by application of Ito's lemmas to $B(t,T) = \exp\left[-r(t,T)(T-t)\right]$,

$$\mathrm{d}B(t,T) = \left[\frac{\partial B}{\partial t} + \frac{\partial B}{\partial r}\mu(r,T) + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\sigma^2(r,T)\right]\mathrm{d}t + \frac{\partial B}{\partial r}\sigma(r,T)\,\mathrm{d}w\,.$$

Equating these two bond price equations, we have:

$$\alpha(r,t,T)B = \left[\frac{\partial B}{\partial t} + \frac{\partial B}{\partial r}\mu(r,T) + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\sigma^2(r,T)\right];$$
$$\beta(r,t,T)B = \frac{\partial B}{\partial r}\sigma(r,T).$$

Now assume that the risk premium is proportional to their returns standard deviation and let the price of risk be a known function of r and time t:

$$\alpha(r,t,T) = r + \lambda(r,t)\frac{1}{B}\frac{\partial B}{\partial r}.$$

Inserted into the bond equation derived above, this leads to

$$rB + \lambda(r,t)\frac{\partial B}{\partial r} = \left[\frac{\partial B}{\partial t} + \frac{\partial B}{\partial r}\mu(r,T) + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\sigma^2(r,T)\right]$$

and finally to the partial differential equation:

$$0 = \frac{\partial B}{\partial t} + \frac{\partial B}{\partial r}[\mu(r,T) - \lambda(r,t)] + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\sigma^2(r,T) - rB; \quad B(r,T,T) = 1.$$

The solution of this equation, although cumbersome, can be determined. For example, if we set the constants $[\mu(r,T) - \lambda(r,t)] = \theta$; $\sigma^2(r,T) = \rho^2$, then the following solution can be verified:

$$B(r,t,T) = \exp\left[-r(T-t) - \frac{1}{2}\theta(T-t)^2 + \frac{1}{6}\rho^2(T-t)^3\right].$$

In general these equations are difficult to solve analytically or numerically and require therefore a certain amount of mathematical and numerical ability. Alternatively, if we set

$$[\mu(r,T) - \lambda(r,t)] = k(\theta - r);\ \sigma^2(r,T) = \rho^2 r.$$

Then we can show that the solution for the bond value is of the affine structure form and therefore given by

$$\ln B(r,t,T) = A(T-t) + rD(T-t).$$

A solution for the function $A(.)$ and $D(.)$ can then be found by substitution.

47.4.4 Default Bonds

There are various models for default-prone bond, falling into one of two categories: structural models and reduced-form models [47.74, 92, 93]. Structural models of default specify a particular value process and assume that default occurs when the value falls below some explicit threshold (for example, default may occur when the debt-to-equity ratio crosses a given threshold). In this sense, default is a *stopping time* defined by the evolution of a representative stochastic process. These models determine both equity and debt prices in a self-consistent manner via arbitrage, or contingent-claims pricing. These models assume often that debt-holders get back a fraction of the face value of the debt, sometimes called the recovery ratio at default. Such an assumption is observed largely in practice with bondholders recovering 20–80% of their investment. This recovery ratio is known a priori, however, in their models. Structural models have a number of additional drawbacks. For example, they cannot incorporate credit-rating changes that occur frequently for default prone (risky) corporate debts. Many corporate bonds undergo credit downgrades by credit-rating agencies before they actually default, and bond prices react to these changes either in anticipation or when they occur. Thus, any valuation model should take into account the uncertainty associated with credit-rating changes as well as the uncertainty surrounding default.

Reduced-form models instead, specify the default process explicitly, interpreting it as an exogenously motivated jump process, usually given as a function of the firm value. This class of models has been investigated for example by *Jarrow* and *Turnbull* [47.92], *Jarrow*, *Lando* and *Turnbull* [47.93, 94], *Duffie* and *Singleton* [47.78], and others. Although these models are useful when fitting default to observed credit spreads, neglecting the underlying value process of the firm renders it less useful when it is necessary to determine credit-spread variations. There are numerous publications regarding default-prone bonds and therefore we only consider some classical and simple examples.

Example 47.3 (structural models): Longstaff–Schwartz [47.95] assume a risk-free interest rate two-factor model with interest rates given by a *Vacicek* [47.96] model. Let V_t and r_t be the time-t values of the firm's assets and the risk-free interest rate, respectively. The dynamics of these two factors is written in terms of the following

equations

$$dV/V = (r-\delta)\,dt + \sigma_1\,dZ_1\,;$$
$$dr = (\alpha - \beta r)\,dt + \sigma_2\,dZ_2\,,$$

where δ, σ_1, α, β and σ_2 are constants, and Z_1 and Z_2, two standard Brownian motion processes with constant correlation coefficient ρ.

In their model, default occurs when the value of the firm declines to a pre-specified boundary (with the par value of the bond – the face amount due on the maturity date – taken as the boundary). As a result, the default boundary is specified exogenously. In the event of default, bondholders recover a constant fraction of the par value of the bond. In the Longstaff–Schwartz model, a risky coupon bond is valued as a simple portfolio of a risky zero-coupon bond whose value for a \$ 1 face value is given by

$$P(V_t\,|F, r_t, T)$$
$$= D(r_t, T)\,[1-(1-w)Q(V_t\,|F, r_t, T)]\,,$$

where $D(.)$ denotes the value of a default-free discount bond given by the *Vacicek* [47.96] model, $Q(.)$ represents the forward default probability while w is the recovery rate.

Example 47.4 (reduced-form models): Failure of structural models to adequately price risky bonds found in the marketplace led to another approach based on reduced-form models of default risk. These make no attempt to define default as an endogenous event (arising from a low level of firm value or cash flow), but rather specify default as an exogenous event and thus do not explicitly incorporate any relationship between leverage and firm value into the model. These models, such as those of *Duffie* and *Singleton* [47.78], *Jarrow* and *Turnbull* [47.92], and *Jarrow*, *Lando* and *Turnbull* [47.93], are based on parameters that can be estimated with readily available data, such as default rates or bond spreads. The model of *Jarrow*, *Lando*, and *Turnbull* ([47.93], for example) assumes that the value of a default-free zero-coupon bond is known at time t. This bond will mature at time T and pay one dollar on maturity. $p(t, T)$ is the value of this bond. If $v_i(t, T)$ denotes the value of a defaultable zero-coupon bond of a firm that currently has credit rating i (for example, AAA) at time t, will mature at time T, and has a promised payoff of \$ 1 at maturity, then Jarrow, Lando, and Turnbull show that:

$$v_i(t, T) = p(t, T)[\phi + (1-\phi)q_i(t, T)]\,,$$

where ϕ is the recovery ratio, the fraction of the face value (\$ 1) that is recovered at time T after default, and $q_i(t, T)$ denotes the probability of a default occurring after T given that the debt has credit rating i as of time t. To arrive at the valuation formula, *Jarrow*, *Lando*, and *Turnbull* [47.93] assume that default is independent of the level of interest rates. However, this assumption is not critical. The independence assumption can be relaxed so that the model of *Jarrow* et al. extended so that the default relaxes the independence assumption and extends the model of Jarrow et al. so that the default probability can depend on the level of interest rates.

47.5 Options

Options are instruments that give the buyer of the option (the long side) the *right to exercise*, for a price, called the *premium*, the delivery of a commodity, a stock, a foreign currency etc. at a given price, called the *strike price*, at *(within)* a given time period, also called the *exercise date*. Such an option is called a *European (American) call* for the buyer. The seller of such an option (the short side), has by contrast the *obligation* to sell the option at the stated strike and exercise date. A *put* option (the long side) provides the option to sell, while for the short seller there is an obligation to buy. There are many types of options however and considerable research on the pricing of options (for example, see [47.24, 32, 97–106]. We shall consider in particular call and put options. Options are traded on many trading floors and mostly, they are defined in a standard manner. Nevertheless, there are also over-the-counter options, which are not traded in specific markets but are used in some contracts to fit specific needs. For example, there are Bermudan and Asian options. The former option provides the right to exercise the option at several specific dates during the option lifetime, while the latter defines the exercise price of the option as an average of the value attained over a certain time interval. Of course, each option, defined in a different way, will lead to alternative valuation formulas. There can be options on real assets, which are not traded but used to define a contract between two parties (real options). The valuation of options has attracted a huge amount of interest and for this reason it

will also be a substantial issue we shall deal within this chapter.

47.5.1 Options Valuation and Martingales

When the underlying price process is a martingale and risk-neutral pricing of financial assets applies, then the price of a cash flow $\tilde{S}_n$ realized at time n is

$$S_0 = \frac{1}{(1+R_f)^n} E^*\left(\tilde{S}_n\right) .$$

Hence the forward price is:

$$V_0 (1+R_f)^n = E^*\left(\tilde{S}_n \,|\Phi_0\right) ,$$

where Φ_0 is a filtration, representing the initial information on the basis of which the expectation is taken (under the risk-neutral distribution where expectation is denoted by $E^*(*)$). If K is the exercise price of a call option for exercise at some time T, then, the price C_t of such an option (as well as a broad variety of other options) under risk-neutral pricing is

$$C_t = \frac{1}{(1+R_f)^{T-t}} E^*\left[\max\left(K - \tilde{S}_T, 0\right)|\Phi_t\right] .$$

A simple example often used is the binomial option model. For simplicity, assume, a stock whose current price is 1 \$ and consider two asset state prices one period hence (H, L), $H > 1$, $L < 1$. Under risk-neutral pricing, then of course $1 = \frac{1}{1+R_f}\left[pH + (1-p)L\right]$, where p is a risk-neutral probability. To determine this probability we construct a replicating portfolio for the call option whose state prices are $(C_H, C_L) = (H-K, 0)$, $H > K$, $L < K$. Let this portfolio consist of the stock and a risk-less zero-coupon bond paying one dollar one period hence and be given initially by $P = a + b$. One period hence, the portfolio state prices are necessarily $(P_H, P_L) = \left[aH + b(1+R_f), aL + b(1+R_f)\right]$. It is a replicating portfolio if $(P_H, P_L) = (C_H, C_L)$. A solution of these replicating asset prices yields both a^* and b^* – the replicating portfolio composition. Since two assets with identical cash flows have the same price, the portfolio price and the call option ought, in complete markets, have the same price and therefore $C = P^* = a^* + b^*$, which provides the desired solution. Since the call option under risk-neutral pricing equals the discounted (at the risk-free rate) value of the call option at its exercise, or:

$$\begin{aligned} C &= \frac{1}{1+R_f}\left[pC_H + (1-p)C_L\right] \\ &= \frac{1}{1+R_f}\left[p\,(H-K)\,,(1-p)\,(0)\right] \end{aligned}$$

we can solve this equation and obtain the risk-neutral probability

$$\begin{aligned} p &= \frac{(1+R_f)\,(1) - L}{H-L} ; \\ q &= 1 - p = \frac{H - (1+R_f)(1)}{H-L} . \end{aligned}$$

This analysis can be repeated for several periods. Explicitly, for an option whose exercise is at time n we obtain by induction

$$\begin{aligned} C &= \frac{1}{(1+R_f)^n} \\ &\quad \times \left[\sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} (H^j L^{n-j} x - K)^+\right] , \\ x &= 1. \end{aligned}$$

We can write this expression in still another form

$$\begin{aligned} C_n &= \frac{1}{(1+r)^n} E^*\left[(x_n - K)^+\right], \quad \text{where} \\ P\left(x_n = H^j L^{n-j} x\right) &= \binom{n}{j} p^j (1-p)^{n-j} . \end{aligned}$$

47.5.2 The Black–Scholes Option Formula

In continuous time and continuous state, the pricing of Black–Scholes options are obtained in a similar manner, albeit using stochastic calculus. The traditional approach is based on the replication of the option value by the construction of a portfolio consisting of the underlying asset (the security) and a risk-free bond. Let $S(t)$ be a security-stock price at time t, distributed as a log-normal process and let V be the value of an asset derived from this stock, which we can write by the following function $V = f(S, t)$, assumed to be differentiable with respect to time and the security-stock $S(t)$. For simplicity, let the security price be given by a log-normal process:

$$\frac{\mathrm{d}S}{S} = \alpha\,\mathrm{d}t + \sigma\,\mathrm{d}W, \quad S(0) = S_0 ,$$

where $\left\{W(t), t \geq 0\right\}$, $W(0) = 0$ is a standard Brownian motion. Let P be a replicating portfolio consisting of bonds and investment in the given stock, $P = B + aS$ or $B = P - aS$, in which case the price of a risk-less bond and the price of a portfolio $P - aS$ is necessarily the same. A perfect hedge is thus constructed by setting: $\mathrm{d}B = \mathrm{d}P - a\,\mathrm{d}S$ where $\mathrm{d}B = R_f B\,\mathrm{d}t$. Now, let $V = C = f(S, t)$ be the option price. Setting the replicating portfolio, we have $P = C$ and $\mathrm{d}P = \mathrm{d}C$, which

is used to obtain a partial differential equation of the option price with appropriate boundary conditions and constraints, providing thereby the solution to the Black–Scholes option price. Each of these steps is translated into mathematical manipulations. First, note that:

$$\mathrm{d}B = \mathrm{d}f - a\,\mathrm{d}S = R_\mathrm{f} B\,\mathrm{d}t\,.$$

By an application of Ito's differential rule we obtain the option price:

$$\begin{aligned}\mathrm{d}C &= \mathrm{d}f \\ &= \left(\frac{\partial f}{\partial t} + \alpha S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2}\right)\mathrm{d}t \\ &\quad + \left(\sigma S\frac{\partial f}{\partial S}\right)\mathrm{d}W\end{aligned}$$

and therefore,

$$\begin{aligned}&\left(\frac{\partial f}{\partial t} + \alpha S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2}\right)\mathrm{d}t \\ &\quad + \left(\sigma S\frac{\partial f}{\partial S}\right)\mathrm{d}W - a\,\mathrm{d}S = R_\mathrm{f}(f - aS)\,\mathrm{d}t\,.\end{aligned}$$

Thereby,

$$\begin{aligned}&\left[\frac{\partial f}{\partial t} + \alpha S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2}\right. \\ &\quad \left. - aS\,(\alpha - R_\mathrm{f}) - R_\mathrm{f} f\right]\mathrm{d}t \\ &\quad + \sigma S\left(-a + \frac{\partial f}{\partial S}\right)\mathrm{d}W = 0\,,\end{aligned}$$

or

$$a = \frac{\partial f}{\partial S};\ \left(\frac{\partial f}{\partial t} + R_\mathrm{f} S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2} - R_\mathrm{f} f\right) = 0$$

and finally

$$\begin{aligned}&-\frac{\partial f}{\partial t} = R_\mathrm{f} S\frac{\partial f}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 f}{\partial S^2} - R_\mathrm{f} f;\quad f(0,t) = 0\,, \\ &\forall t \in [0,T]\,,\qquad f(S,T) = \max\,[0,\,S(T) - K]\,.\end{aligned}$$

The boundary conditions are specified by the fact that the option cannot be exercised until the exercise time (unlike an American option, as we shall see below) and therefore it is worthless until that time. At the exercise date T however, it equals $f(S,T) = \max[0,\,S(T) - K]$. The solution was shown by Black and Scholes to be

$$W = f(S,t) = S\Phi(d_1) - K\,\mathrm{e}^{-R_\mathrm{f} t}\Phi(\mathrm{d}_2)\,,$$

where

$$\begin{aligned}&\Phi(y) = (2\pi)^{-1/2}\int_{-\infty}^{y}\mathrm{e}^{-u^2/2}\,\mathrm{d}u\,; \\ &d_1 = \left(\frac{\log(S/K) + (T-t)(R_\mathrm{f} + \sigma^2/2)}{\sigma\sqrt{T-t}}\right)\,; \\ &d_2 = d_1 - \sigma\sqrt{T-t}\,.\end{aligned}$$

This result is remarkably robust and holds under very broad price processes. Further, it can be estimated by simulation very simply. There are many computer programs that compute these option prices as well as their sensitivities. The price of a put option is calculated in a similar manner (see also [47.107, 108]).

47.5.3 Put–Call Parity

The put–call parity relationship establishes a relationship between the price of a put and that of a call. It can be derived by a simple arbitrage argument between two equivalent portfolios, yielding the same payoff regardless of the stock price. Their value must therefore be the same. To do so, construct the following two portfolios at time t:

Time t	Time T
	$S_T < K$
(1) $c + K\,\mathrm{e}^{-R_\mathrm{f}(T-t)}$	K
(2) $p + S_t$	$K = (K - S_T) + S_T$

Time t	Time T
	$S_T > K$
(1) $c + K\,\mathrm{e}^{-R_\mathrm{f}(T-t)}$	$(S_T - K) + K = S_T$
(2) $p + S_t$	S_T

We see that at time T, the two portfolios yield the same payoff $\max(S_T, X)$ which implies the same price at time t. Thus

$$c + K\,\mathrm{e}^{-R_\mathrm{f}(T-t)} = p + S_t\,.$$

If this is not the case then there would be some arbitrage opportunity. In this sense, computing European options prices is simplified, since knowing one leads necessarily to knowing the other.

When we consider dividend-paying options, the put–call parity relationships are slightly altered. Let D denote the present value of the dividend payments during the

lifetime of the option (occurring at the time of its ex-dividend date), then:

$$c > S - D - K\,\mathrm{e}^{-R_\mathrm{f}(T-t)}\,,$$
$$p > D + K\,\mathrm{e}^{-R_\mathrm{f}(T-t)} - S\,.$$

Similarly, for put–call parity in a dividend-paying option, we have the following bounds

$$S - D - K < C - P < S - K\,\mathrm{e}^{-R_\mathrm{f}(T-t)}\,.$$

Put–call parity can be applied similarly between securities denominated in different currencies. For example, let α be the euro/dollar exchange rate (discounted at the dollar risk-free rate) and let $R_{\mathrm{f,E}}$ be the euro-area discount rate. Then, by put–call parity, we have

$$c + \frac{K}{1 + R_{\mathrm{f,E}}} = p + \alpha\,,$$

which can be used as a regression equation to determine the actual exchange rate based on options data on currencies exchange.

47.5.4 American Options – A Put Option

American options, unlike European options, may be exercised prior to the expiration date. The price of such options is formulated in terms of stochastic dynamic programming arguments. As long as the option is *alive* we may either exercise it or maintain it, continuing to hold it. In a continuation region, the value of the option is larger than the value of its exercise and therefore, it is optimal to wait. In the exercise region, it is optimal to exercise the option and cash in the profits. If the time to the option's expiration date is t, then the exercise of the option provides a profit $K - S(t)$. In this latter case, the exercise time is a *stopping time*, and the problem is terminated. Another way to express such a statement using dynamic programming arguments is:

$$f(S,t) = \max\left[K - S(t),\, \mathrm{e}^{-R_\mathrm{f}\,\mathrm{d}t} E f(S + \mathrm{d}S, t + \mathrm{d}t)\right]\,,$$

where $f(S,t)$ is the option price at time t when the underlying stock price is S and one of the two alternatives holds at equality. At the contracted strike time of the option, we have necessarily, $f(S,0) = K - S(0)$. The solution of the option's exercise time is difficult however and has generated a large number of studies seeking to solve the problem analytically or numerically. Noting that the solution is of barrier type, meaning that there is some barrier $X^*(t)$ that separates the exercise and continuation regions, we have

$$\begin{cases} \text{If } K - S(t) \geq X^*(t) & \text{exercise region:} \\ & \text{stopping time} \\ K - S(t) < X^*(t) & \text{continuation region}\,. \end{cases}$$

The solution of the American put problem consists then of selecting the optimal exercise barrier [47.109, 110]. A number of studies have attempted to do so, including [47.111] as well as many other authors. Although the analytical solutions of American put options are hard to reach, there are some problems that have been solved analytically. For most practical problems, numerical and simulation techniques are used.

Explicitly, assume that an American put option derived from a security is exercised at time $\tau < T$ where T is the option exercise period while the option exercise price is K. Let the underlying stock price be a risk-neutral process:

$$\frac{\mathrm{d}S(t)}{S(t)} = R_\mathrm{f}\,\mathrm{d}t + \sigma\,\mathrm{d}W(t),\; S(0) = S_0\,.$$

Under risk-neutral pricing, the value of the option equals the discounted value (at the risk-free rate) at the optimal exercise time $\tau^* < T$, namely:

$$J(S,T) = \max_{\tau \leq T} E_S\,\mathrm{e}^{-R_\mathrm{f}\tau}\,[K - S(\tau), 0]\,.$$

Thus,

$$J(S,t) = \begin{cases} K - S(t) & \text{exercise region: stopping time} \\ \mathrm{e}^{-R_\mathrm{f}\,\mathrm{d}t} E J(S + \mathrm{d}S, t + \mathrm{d}t) & \\ & \text{continuation region}\,. \end{cases}$$

In the continuation region we have explicitly:

$$\begin{aligned} J(S,t) &= \mathrm{e}^{-R_\mathrm{f}\,\mathrm{d}t} E J(S + \mathrm{d}S, t + \mathrm{d}t) \\ &\approx (1 - R_\mathrm{f}\,\mathrm{d}t) \\ &\quad \times E\left[J(S,t) + \frac{\partial J}{\partial t}\,\mathrm{d}t + \frac{\partial J}{\partial S}\,\mathrm{d}S + \frac{1}{2}\frac{\partial^2 J}{\partial S^2}(\mathrm{d}S)^2\right] \end{aligned}$$

which is reduced to the following partial differential equation

$$-\frac{\partial J}{\partial t} = -R_\mathrm{f} J(S,t) + \frac{\partial J}{\partial S} R_\mathrm{f} S + \frac{1}{2}\frac{\partial^2 J}{\partial S^2}\sigma^2 S^2\,,$$

while in the exercise region:

$$J(S,t) = K - S(t)\,.$$

For a perpetual option, note that the option price is not a function of time but of price only and therefore $\frac{\partial J}{\partial t} = 0$ and the option price is given by an ordinary differential equation of second order

$$0 = -R_f J(S) + \frac{dJ}{dS} R_f S + \frac{1}{2} \frac{d^2 J}{dS^2} \sigma^2 S^2 .$$

Assume that an interior solution exists, with an exercise at price S^*, $S(t) \le S^*$, $S^* \le K$. These specify the two boundary conditions required to solve our equation. In the exercise region $J(S^*) = K - S^*$, while for optimal exercise price $\frac{dJ(S)}{dS}\big|_{S=S^*} = -1$. Let the solution be of the type $J(S) = qS^{-\lambda}$. This reduces the differential equation to an equation we solve for $\lambda : \sigma^2 \frac{\lambda(\lambda+1)}{2} - \lambda R_f - R_f = 0$ and $\lambda^* = 2R_f/\sigma^2$. At the exercise boundary S^* however: $J(S^*) = qS^{*-\lambda^*} = K - S^*$; $dJ(S^*)/dS^* = -\lambda^* q S^{*-\lambda^*-1} = -1$. These two equations are solved for q and $S^*[0, 1]$ leading to: $S^* = \lambda^* K/(1+\lambda^*)$ and $q = (\lambda^*)^{\lambda^*} K^{1+\lambda^*}/(1+\lambda^*)^{1+\lambda^*}$ and the option price is thus:

$$J(S) = \left(\frac{(\lambda^*)^{\lambda^*} K^{1+\lambda^*}}{(1+\lambda^*)^{1+\lambda^*}} \right) S^{-\lambda^*} ,$$

$$\lambda^* = \frac{2R_f}{\sigma^2}, \quad S^* = \frac{\lambda^*}{1+\lambda^*} K .$$

Thus the solution of the perpetual American put is explicitly given by:

$$\begin{cases} \text{Sell if} & S \le S^* \\ \text{Hold if} & S > S^* . \end{cases}$$

When the option time is finite, say T, this problem is much more difficult to solve however. Further, for an American call, it is easily demonstrated that it equals in fact the price of the European call.

In discrete time, a similar approach may be applied if risk-neutral pricing can be applied. For example, consider again the binomial option model considered earlier. The stock can assume at time n the following prices: $H^i L^{n-i} x - K, i = 0, 1, 2, \ldots n+1$, where p is the probability of the price increasing (and $1-p$, the probability that it decreases) and x is the initial price (at time $t = 0$). The price of a put option with an option maturity at time n is then: $P_n(i) = \max(K - H^i L^{n-i} x, 0)$. Suppose that at time t the put is exercised, then the profit is $P_t(i)$. Alternatively, say that the option is not sold at t. In this case, by risk-neutral pricing, the price of the option is

$$P^*_{t+1}(i) = \frac{1}{1+R_f} \left[pP_{t+1}(i+1) + (1-p) P_{t+1}(i) \right] .$$

Thus, by the recurrence (Bellman) equation for this problem, we have:

$$P_t(i) = \max \left[K - H^i L^{n-i} x,\ P^*_{t+1}(i) \right]$$

with boundary condition

$$P_n(i) = \max \left(K - H^i L^{n-i} x, 0 \right) ,$$

and a solution can be found by numerical techniques.

47.5.5 Departures from the Black–Scholes Equation

Any departure from the basic assumptions underlying the Black–Scholes model will necessarily alter the Black–Scholes (BS) solution. For example, if volatility is stochastic, if interest rates are stochastic, if stock prices are not log-normal, etc. the solution will not be necessarily a BS solution. For many cases however, it is possible to construct replicating portfolios and thereby remain within the assumptions that markets are complete. Below we shall consider a number of such cases to demonstrate how we might proceed in different manners. These approaches however, are based on a valuation based on risk-neutral pricing (for example, *Hull* [47.32], *Jarrow* and *Rudd* [47.112]).

The BS option price depends, of course, on the assumptions made regarding the underlying price process. Further, it depends essentially on the stock volatility, which cannot be observed directly. For this reason, the relationships between the option price and volatility have been taken to reflect one or the other. In other words, given the options price and other observables (interest rates, strike price, etc.), the implied volatility is that volatility that solves the BS price equation: $\hat{C} = W(.|\sigma_{\text{imp}})$, where $\hat{C}$ is the current option price and $C = W(.|\sigma)$ is the theoretical option price with an implied volatility $\sigma = \sigma_{\text{imp}}$. Importantly, when the volatility is constant then σ_{imp} does not change as a function of T and K and it equals the true historical volatility. However, in practice when we calculate this implied volatility as a function of (T, K), we observe that there are some variations and therefore the BS model cannot be considered as the true market option price. Further, when the underlying price changes, the implied volatility can be a function of time as well and as a result, the implied volatility is a function $\sigma_{\text{imp}}(t)$. When we consider the options price variations as a function of the strike K, we observe a volatility skew which is the well-known volatility smile. Skewness is smaller however for at-the-money options (in which case, the BS model is

a good predictor of option price). The valuation of options in these circumstances is more difficult and there are, commensurably, numerous studies and extensions that calculate option prices. For example, [47.113, 114] consider transaction costs, [47.90, 115–117] consider option prices with stochastic volatility. *Nelson* and *Ramaswamy* [47.118] use discretized approximations and, even in physics, option pricing is considered as an application [47.119]. We consider below some well-known cases.

Option Valuation and Stochastic Volatility

When the underlying process has a stochastic volatility the replication of an option price by a portfolio requires special attention. We may proceed then by finding an additional asset to use (for example, another option with different maturity and strike price). Consider the following stochastic volatility process as an example

$$\begin{aligned} \mathrm{d}p/p &= \alpha\,\mathrm{d}t + \sqrt{V}\,\mathrm{d}w, & p(0) &= p_0 \\ \mathrm{d}V/V &= \mu\,\mathrm{d}t + \xi\,\mathrm{d}z, & V(0) &= v_0\,; \\ E(\mathrm{d}w\,\mathrm{d}z) &= \rho\,\mathrm{d}t\,, \end{aligned}$$

where (w, z) are two Brownian motions with correlation ρ. A call option would in this case be a function of both p and V, as we saw earlier for the BS option model, or $C(t, p, V)$. Application of Ito's lemma yields

$$\begin{aligned} \mathrm{d}C = \Bigg[& \frac{\partial C}{\partial t} + \frac{\partial C}{\partial p}\alpha p + \frac{\partial C}{\partial V}\mu V + \frac{1}{2}\frac{\partial^2 C}{\partial p^2}(\sigma p)^2 \\ & + \frac{1}{2}\frac{\partial^2 C}{\partial V^2}(\xi V)^2 + \frac{\partial^2 C}{\partial p \partial V}(\mu V + \rho\xi\sigma p V)\Bigg]\mathrm{d}t \\ & + \frac{\partial C}{\partial p}(\sigma p\,\mathrm{d}W) + \frac{\partial C}{\partial V}(\xi V\,\mathrm{d}Z)\,. \end{aligned}$$

The first term in the brackets is the deterministic component while the remaining ones are stochastic terms that ought to be nullified by an appropriate portfolio (i. e. hedged) if we are to apply a risk-neutral framework. Since there are two sources of risk, we require two assets in addition to the underlying asset price. For this reason, we construct a replication portfolio by: $X = n_1 p + n_2 C_2 + B$, $B = (X - n_1 p - n_2 C_2)$, where n_1, n_2 are the number of stock shares and another option with different maturity. In this case, proceeding as we have for the BS model, we have $\mathrm{d}C_1 = \mathrm{d}X$ and therefore

$$\begin{aligned} & \mathrm{d}C_1 - n_1\,\mathrm{d}p - n_2\,\mathrm{d}C_2 = rB\,\mathrm{d}t \\ & = r(C_1 - n_1 p - n_2 C_2)\mathrm{d}t \quad \text{or} \\ & (\mathrm{d}C_1 - rC_1\,\mathrm{d}t) - n_1\,(\mathrm{d}p - rp\,\mathrm{d}t) \\ & - n_2\,(\mathrm{d}C_2 - rC_2\,\mathrm{d}t) = 0 \end{aligned}$$

which provides the equations needed to determine a hedging portfolio given by

$$\begin{aligned} & \mathrm{d}\Phi_1 = (\mathrm{d}C_1 - rC_1\,\mathrm{d}t)\,, \quad \text{hence} \quad \Phi_1 = \mathrm{e}^{-rt}C_1\,; \\ & \mathrm{d}\Phi_2 = (\mathrm{d}p - rp\,\mathrm{d}t)\,, \quad \text{hence} \quad \Phi_2 = \mathrm{e}^{-rt}p\,, \\ & \mathrm{d}\Phi_3 = (\mathrm{d}C_2 - rC_2\,\mathrm{d}t)\,, \quad \text{hence} \quad \Phi_2 = \mathrm{e}^{-rt}C_2\,; \\ & \mathrm{d}\Phi_1 = n_1\,\mathrm{d}\Phi_2 + n_2\,\mathrm{d}\Phi_3\,. \end{aligned}$$

As a result,

$$\mathrm{d}\left(\mathrm{e}^{-rt}C_1\right) = n_1\,\mathrm{d}\left(\mathrm{e}^{-rt}p\right) + n_2\,\mathrm{d}\left(\mathrm{e}^{-rt}C_2\right)\,.$$

Further,

$$\begin{aligned} & \frac{\mathrm{d}C_1}{C_1} = \mu_1\,\mathrm{d}t + \sigma_1\,\mathrm{d}W_1, \quad \text{with} \quad \lambda_1 = \frac{(\mu_1 - r)}{\sigma_1}\,, \\ & \text{hence} \quad \frac{\mathrm{d}C_1}{C_1} - r\,\mathrm{d}t = \sigma_1\,(\lambda_1\,\mathrm{d}t + \mathrm{d}W_1) = \sigma_1\,\mathrm{d}\tilde{W}_1 \end{aligned}$$

while $(\lambda_1\,\mathrm{d}t + \mathrm{d}W_1) = \mathrm{d}\tilde{W}_1$, is the risk-neutral measure. If we apply a CAPM risk valuation, we have then:

$$\begin{aligned} \frac{1}{\mathrm{d}t}E\left(\frac{\mathrm{d}C_1}{C_1} - r\right) &= \sigma_1\lambda_1 \\ &= \left[\left(R_p - r\right)\beta_{cp} + (R_V - r)\,\beta_{cV}\right], \end{aligned}$$

where R_p is the stock mean return, $\beta_{cp} = \frac{p}{C_1}\frac{\partial C_1}{\partial p}\beta_p$ is the stock beta, R_V is the volatility drift while $\beta_{cV} = \frac{V}{C_1}\frac{\partial C_1}{\partial V}\beta_V$ is the beta due to volatility. We therefore obtain the following equations:

$$\begin{aligned} & \frac{1}{\mathrm{d}t}E\left(\frac{\mathrm{d}C_1}{C_1}\right) \\ & = r + \left[(\alpha - r)\,\frac{p}{C_1}\frac{\partial C_1}{\partial p}\beta_p + (\mu - r)\,\beta_V\frac{V}{C_1}\frac{\partial C_1}{\partial V}\right]; \\ & R_p = \alpha; \quad R_V = \mu, \quad \lambda_V = (\mu - r)\,V\beta_V\,, \end{aligned}$$

where λ_V is the risk premium associated with the volatility. Thus,

$$\frac{1}{\mathrm{d}t}E\left(\frac{\mathrm{d}C_1}{C_1}\right) = r + \left[(\alpha - r)\,\frac{p}{C_1}\frac{\partial C_1}{\partial p} + \frac{\lambda_V}{C_1}\frac{\partial C_1}{\partial V}\right]$$

which we equate to the option we are to value. Since,

$$\frac{1}{\mathrm{d}t}E\left(\frac{\mathrm{d}C}{C}\right) = r + \left[(\alpha - r)\,\frac{p}{C}\frac{\partial C}{\partial p} + \frac{\lambda_V}{C}\frac{\partial C}{\partial V}\right]$$

and obtain at last:

$$\begin{aligned} & \frac{1}{\mathrm{d}t}E\left(\frac{\mathrm{d}C}{C}\right) \\ & = \frac{1}{\mathrm{d}t}E\left[\frac{\partial C}{C\partial t}\mathrm{d}t + \frac{\partial C}{C\partial p}\mathrm{d}p + \frac{\partial C}{C\partial V}\mathrm{d}V\right. \\ & \quad + \frac{1}{2}\frac{\partial^2 C}{C\partial p^2}(\mathrm{d}p)^2 + \frac{1}{2}\frac{\partial^2 C}{C\partial V^2}(\mathrm{d}V)^2 \\ & \quad \left. + \frac{\partial^2 C}{C\partial p\partial V}(\mathrm{d}p\,\mathrm{d}V)\right] \end{aligned}$$

which leads to a partial differential equation we might be able to solve numerically. Or:

$$E\left[\frac{\partial C}{\partial t}\,\mathrm{d}t+\frac{\partial C}{\partial p}\,\mathrm{d}p+\frac{\partial C}{\partial V}\,\mathrm{d}V+\frac{1}{2}\frac{\partial^2 C}{\partial p^2}\,(\mathrm{d}p)^2 +\frac{1}{2}\frac{\partial^2 C}{\partial V^2}\,(\mathrm{d}V)^2+\frac{\partial^2 C}{\partial p\partial V}\,(\mathrm{d}p\,\mathrm{d}V)\right] =rC+\left[(\alpha-r)\,p\frac{\partial C}{\partial p}+\lambda_V\frac{\partial C}{\partial V}\right]$$

and explicitly,

$$\left[\frac{\partial C}{\partial t}+\frac{\partial C}{\partial p}rp+\frac{\partial C}{\partial V}\,(\mu V-\lambda_V)+\frac{1}{2}\frac{\partial^2 C}{\partial p^2}Vp^2 +\frac{1}{2}\frac{\partial^2 C}{\partial V^2}V^2\xi^2+\frac{\partial^2 C}{\partial p\partial V}pV^{3/2}\xi\rho-rC\right]=0\,,$$

where $\lambda_V=(\mu-r)V\beta_V$, as stated earlier. Of course the boundary constraints are then $C(T,\,p)=\max(p-K,0)$. The analytical treatment of such problems is clearly unlikely however (see also [47.120].

Options and Jump Processes [47.121]

The valuation of an option with a jump price process also involves two sources of risk, the diffusion and the jump. Merton considered such a problem for the following price process:

$$\frac{\mathrm{d}p}{p}=\alpha\,\mathrm{d}t+\sigma\,\mathrm{d}w+K\,\mathrm{d}Q\,,$$

where $\mathrm{d}Q$ is an adapted Poisson process with parameter $q\Delta t$. In other words, $Q(t+\Delta t)-Q(t)$ has a Poisson distribution function with mean $q\Delta t$ or for infinitesimal time intervals

$$\mathrm{d}Q=\begin{cases}1 & \text{w. p.}\quad q\,\mathrm{d}t\\ 0 & \text{w. p.}\quad (1-q)\,\mathrm{d}t\,.\end{cases}$$

Let $F=F(p,t)$ be the option price. When a jump occurs, the new option price is $F[p(1+K)]$. As a result,

$$\mathrm{d}F=\{F[p(1+K)]-F\}\,\mathrm{d}Q$$

when no jump occurs, we have

$$\mathrm{d}F=\frac{\partial F}{\partial t}\,\mathrm{d}t+\frac{\partial F}{\partial p}\,\mathrm{d}p+\frac{1}{2}\frac{\partial^2 F}{\partial p^2}(\mathrm{d}p)^2$$

and explicitly, letting $\tau=T-t$ be the remaining time to the exercise date, we have

$$\mathrm{d}F=\left(-\frac{\partial F}{\partial\tau}+\alpha p\frac{\partial F}{\partial p}+\frac{1}{2}p^2\sigma^2\frac{\partial^2 F}{\partial p^2}\right)\mathrm{d}t +p\sigma\frac{\partial F}{\partial p}\,\mathrm{d}w\,.$$

Combining these two equations, we obtain

$$\mathrm{d}F=a\,\mathrm{d}t+b\,\mathrm{d}w+c\,\mathrm{d}Q\,,$$
$$a=\left(-\frac{\partial F}{\partial\tau}+\alpha p\frac{\partial F}{\partial p}+\frac{1}{2}p^2\sigma^2\frac{\partial^2 F}{\partial p^2}\right);$$
$$b=p\sigma\frac{\partial F}{\partial p};\quad c=F[p(1+K)]-F$$

with

$$E(\mathrm{d}F)=(a+qc)\,\mathrm{d}t\quad\text{since}\quad E(\mathrm{d}Q)=q\,\mathrm{d}t\,.$$

To eliminate the stochastic elements (and thereby the risks implied) in this equation, we shall construct a portfolio consisting of the option and a stock. To eliminate the *Wiener risk*, i.e. the effect of "$\mathrm{d}w$", we let the portfolio Z consist of a future contract whose price is p, for which a proportion v of stock options is sold (which will be calculated such that this risk disappears). In this case, the value of the portfolio is

$$\mathrm{d}Z=p\alpha\,\mathrm{d}t+p\sigma\,\mathrm{d}w+pK\,\mathrm{d}Q -(va\,\mathrm{d}t+vb\,\mathrm{d}w+vc\,\mathrm{d}Q)\,.$$

If we set $v=p\sigma/b$ and insert in the equation above (as done by Black–Scholes), then we will eliminate the Wiener risk since:

$$\mathrm{d}Z=p(\alpha-\sigma a/b)\,\mathrm{d}t+(p\sigma-vb)\,\mathrm{d}w +p(K-\sigma c/b)\,\mathrm{d}Q$$

or

$$\mathrm{d}Z=p(\alpha-\sigma a/b)\,\mathrm{d}t+p(K-\sigma c/b)\,\mathrm{d}Q\,.$$

In this case, if there is no jump, the evolution of the portfolio follows the differential equation

$$\mathrm{d}Z=p(\alpha-\sigma a/b)\,\mathrm{d}t\quad\text{if there is no jump}\,.$$

However, if there is a jump, then the portfolio evolution is

$$\mathrm{d}Z=p(\alpha-\sigma a/b)\,\mathrm{d}t+p(K-\sigma c/b)\,.$$

Since the jump probability equals $q\,\mathrm{d}t$, we obviously have

$$\frac{E(\mathrm{d}Z)}{\mathrm{d}t}=p(\alpha-\sigma a/b)+pq(K-\sigma c/b)\,.$$

There remains a risk in the portfolio due to the jump. To eliminate it we can construct another portfolio using an option F' (with exercise price E') and a future contract such that the terms in $\mathrm{d}Q$ are eliminated as well. Then, constructing a combination of the first (Z) portfolio and the second portfolio (Z'), both sources of uncertainty

will be reduced. Applying the arbitrage argument (stating that there cannot be a return to a risk-less portfolio which is greater than the risk-less rate of return r) we obtain the proper proportions of the risk-less portfolio.

Alternatively, finance theory [and in particular, application of the capital asset pricing model (CAPM)] state that any risky portfolio has a rate of return in a small time interval $\mathrm{d}t$ which is equal the risk-less rate r plus a return premium for the risk assumed, which is proportional to its effect. Thus, using the CAPM we can write

$$E\frac{\mathrm{d}Z}{Z\,\mathrm{d}t} = r + \lambda\frac{p(K-\sigma c/b)}{Z}\,,$$

where λ is assumed to be a constant and expresses the market price for the risk associated with a jump. This equation can be analyzed further, leading to the following partial differential equation which remains to be solved (once the boundary conditions are specified):

$$-\frac{\partial F}{\partial \tau} + \left((\lambda - q)\left\{pK\frac{\partial F}{\partial p} - F[p(1+K)-F]\right\}\right) + \frac{1}{2}\frac{\partial^2 F}{\partial p^2}p^2\sigma^2 - rF = 0$$

with boundary condition

$$F(T) = \max\,[0,\, p(T) - E]\ .$$

Of course, for an American option, it is necessary to specify the right to exercise the option prior to its final exercise date, or

$$F(t) = \max\left[F^*(t),\, p(t) - E\right]\,,$$

where $F^*(t)$ is the value of the option which is not exercised at time t and given by the solution of the equation above. The solution of this equation is of course much more difficult than the Black–Scholes partial differential equation. Additional papers and extensions include for example, [47.122–124] as well as [47.125, 126].

Call Options on Bonds

Options on bonds are popular products traded in many financial markets. To value these options requires both an interest rate model and a term-structure bond price process. The latter is needed to construct the evolution over time of the underlying bond (say a T bond), which confers the right to exercise it at time $S < T$, in other words, the bond value at time S, whose value is given by an S-bond. To do so, we proceed in two steps: first we evaluate the term structure for a T and an S bond and then proceed to determine the value of a T bond at time S, which is used to replace the spot price at time S in the plain option model of Black–Scholes.

First we construct a hedging portfolio consisting of the two maturities S and T bonds ($S < T$). This portfolio will provide a synthetic rate, equated to the spot interest rate so that no arbitrage is possible. We denote by $k(t)$ this synthetic rate. For example, let the interest process

$$\mathrm{d}r = \mu(r,t)\,\mathrm{d}t + \sigma(r,t)\,\mathrm{d}W$$

and construct a portfolio of these two bonds, whose value is V, with:

$$\frac{\mathrm{d}V}{V} = n_S\frac{\mathrm{d}B(t,S)}{B(t,S)} + n_T\frac{\mathrm{d}B(t,T)}{B(t,T)}\,.$$

The T and S bond values are however, given by:

$$\frac{\mathrm{d}B(t,T)}{B(t,T)} = \alpha_T(r,t)\,\mathrm{d}t + \beta_T(r,t)\,\mathrm{d}W\,,$$

where as seen earlier in the previous section, the term structure is

$$\alpha_T(r,t) = \frac{1}{B(t,T)}\left[\frac{\partial B}{\partial t} + \frac{\partial B}{\partial r}\mu(r,T) + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\sigma^2(r,T)\right];$$
$$\beta_T(r,t) = \frac{1}{B(t,T)}\frac{\partial B}{\partial r}\sigma(r,T)\,.$$

Similarly, for an S-Bond,

$$\frac{\mathrm{d}B(t,S)}{B(t,S)} = \alpha_S(r,t)\,\mathrm{d}t + \beta_S(r,t)\,\mathrm{d}W$$

with

$$\alpha_S(r,t) = \frac{1}{B(t,S)}\left[\frac{\partial B}{\partial t} + \frac{\partial B}{\partial r}\mu(r,S) + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\sigma^2(r,S)\right];$$
$$\beta_S(r,t) = \frac{1}{B(t,S)}\frac{\partial B}{\partial r}\sigma(r,S)\,.$$

Replacing the terms for the mean rate of growth in the bond value and its diffusion, we have

$$\frac{\mathrm{d}V}{V} = (n_S\alpha_S + n_T\alpha_T)\,\mathrm{d}t + (n_S\beta_S + n_T\beta_T)\,\mathrm{d}W\,.$$

For a risk-less portfolio we require that the portfolio volatility be null. Further, since initially the portfolio was worth only one dollar, we obtain two equations in two unknowns (the portfolio composition), which we can solve

$$\begin{cases} n_S\beta_S + n_T\beta_T = 0 \\ n_S + n_T = 1 \end{cases} \Rightarrow \begin{cases} n_S = \frac{\beta_T}{\beta_T - \beta_S} \\ n_T = -\frac{\beta_S}{\beta_T - \beta_S}\,. \end{cases}$$

The risk-less portfolio thus has a rate of growth which we call the synthetic rate, or

$$\frac{\mathrm{d}V}{V} = \left(\frac{\beta_T\alpha_S - \beta_S\alpha_T}{\beta_T - \beta_S}\right)\mathrm{d}t = k(t)\mathrm{d}t\,.$$

This rate is equated to the spot rate as stated above, providing thereby the following equality:

$$\left(k(t) = \frac{\beta_T\alpha_S - \beta_S\alpha_T}{\beta_T - \beta_S}\right) \Rightarrow k(t) = r(t) \quad \text{or}$$
$$\frac{r(t)-\alpha_S}{\beta_S} = \frac{r(t)-\alpha_T}{\beta_T} = \lambda(t)$$

with $\lambda(t)$ the price of risk per unit volatility. Each bond with maturity T and S has at its exercise time a one dollar denomination, the value of each of these (S and T) bonds is given by

$$0 = \frac{\partial B_T}{\partial t} + \frac{\partial B_T}{\partial r}\left[\mu(r,T) - \lambda\beta_T\right] + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\beta_T^2 - rB_T\,;$$
$$B(r,T) = 1\,,$$
$$0 = \frac{\partial B_S}{\partial t} + \frac{\partial B_S}{\partial r}\left[\mu(r,S) - \lambda\beta_S\right] + \frac{1}{2}\frac{\partial^2 B}{\partial r^2}\beta_S^2 - rB_S\,;$$
$$B(r,S) = 1\,.$$

Given a solution to these two equations, we define the option value of a call on a T bond with $S < T$ and strike price K, to be:

$$X = \max\left[B(S,T) - K, 0\right]$$

where $B(S,T)$ is the price of the T bond at time S. $B(S,T)$ is of course found by solving for the term structure and then equating $B(r,S,T) = B(S,T)$. To simplify matters, say that the solution (valued at time t) for the T bond is given by $F(t,r,T)$, then at time S, this value is $F(S,r,T)$, to which we equate $B(S,T)$. In other words,

$$X = \max\left[F(S,r,T) - K, 0\right]\,.$$

Now, if the option price is $P(.)$, then as we have seen in the plain vanilla model in the previous chapter, the value of the bond is found by solving for P in the following partial differential equation

$$0 = \frac{\partial P}{\partial t} + \mu(r,t)\frac{\partial P}{\partial r} + \sigma^2\frac{1}{2}\frac{\partial^2 P}{\partial r^2} - rP\,;$$
$$P(S,r) = \max\left[F(S,r,T) - K, 0\right]\,.$$

Although this might be a difficult problem to solve numerically, there are mathematical tools that allow the finding of such solution. A special case of interest consists of using the term structure model in the problem above, also called the affine term structure (ATS) model, which was indicated earlier in the previous section. In this case, we have:

$$F(t,r,T) = \mathrm{e}^{A(t,T) - rD(t,T)}\,,$$

where $A(.)$ and $D(.)$ are calculated by the term structure model while the option valuation model becomes

$$0 = \frac{\partial P}{\partial t} + \mu(r,t)\frac{\partial P}{\partial r} + \sigma^2\frac{1}{2}\frac{\partial^2 P}{\partial r^2} - rP\,;$$
$$P(S,r) = \max\left[\mathrm{e}^{A(S,T) - rD(S,T)} - K, 0\right]\,.$$

Again, these problems are mostly solved by numerical or simulation techniques.

47.6 Incomplete Markets and Implied Risk-Neutral Distributions

Markets are incomplete when we cannot generate any random cash flow by an appropriate portfolio strategy. The market is then deemed *not rich enough*. Technically, this may mean that the number of assets that make up a portfolio is smaller than the number of risk sources plus one. In the Arrow–Debreu framework seen earlier, this corresponds to rank condition $\boldsymbol{D} \equiv M$, providing a unique solution to the linear pricing equation. If markets are not complete or close to it, financial markets cannot uniquely value assets and there may be opportunities for arbitrage. In such circumstances, financial markets may be perceived as too risky, perhaps *chaotic* and therefore profits may be too volatile, the risk premium would then be too high and investment horizons smaller, thereby reducing investments. Finally, contingent claims may have an infinite number of prices (or equivalently an infinite number of martingale measures). As a result, valuation becomes, forcibly, utility based or based on some other mechanism, which is *subjective* rather than based on the market mechanism. *Ross* [47.104] has pointed out that

> *it is a truism that markets are not complete in the obvious sense that there exist contingencies that have no clearly associated market prices, but, it is not always immediately clear how meaningful this is*

for either pricing or efficiency. Some contingencies may have no markets but may be so trivial as to be insurable in the sense that their associated events are small and independent of the rest of the economy, others, may be replicable while not directly traded

Thus, even if markets are incomplete, we may be able to determine some mechanism which will still allow an approach to asset pricing. Of course, this will require the exact sense of the market incompleteness and determine a procedure to complete it. Earlier, we pointed out some sources of incompleteness, but these are not the only ones. It may arise because of lack of liquidity (leading to market-makers bid/ask spreads for which trading micromodels are constructed); it may be due to excessive friction defined in terms of taxes, indivisibility of assets, varying rates for lending and borrowing (such as no short sales and various portfolio constraints); it could be due to transaction costs and to information asymmetries (insider trading, leading to mis-pricing) indicating that one dimension along which markets are clearly incomplete is that of time. Most traded derivatives markets – futures and options – extend only a few years at most in time and, even when they are formally quoted further out, there is generally little or no liquidity in the far contracts. Yet, it is becoming increasingly common in the world of derivatives to be faced with long-run commitments while liquid markets only provide trading opportunities over shorter run horizons. These are by no means the only situations that lead to incomplete markets. Choice – too much or too little of it – may also induce incompleteness. Rationality implies selecting the best alternative but, when there are too many or the search cost is too high, often investors seek "satisficing solutions" (in the sense of Herbert A. Simon). Barry Schwartz in an article in Scientific American (April 2004) points out, for example, that too much choice may induce an ill feeling and therefore to suboptimal decisions. In addition, regrets [47.17, 36, 37, 127–134], search and other costs can also affect investors' rationality (in the sense of finance's fundamental theory). The Financial Times has pointed out that some investment funds seek to capitalize on human frailties to make money. For example: are financial managers human? Are they always rational, mimicking Star Trek's Mr. Spock? Are they devoid of emotions and irrationality? Psychological decision-making processes integrated in economic rationales have raised serious concerns regarding the rationality axioms of decision-making (DM) processes. There are of course, many challenges to reckon with in understanding human behavior. Some of these include: thought processes based on decision making-approaches focusing on the one hand on the big pictures versus compartmentalization; the effects of under- versus overconfidence on decision making; the application of heuristics of various sorts applied in trading and DM processes. These heuristics are usually based on simple rules. In general, the violation of the assumptions made regarding the definition of rational decision makers and decision makers' psychology are very important issues to reckon with when asset prices in incomplete markets are to be defined (some related references include [47.87, 133, 135–143] and [47.144]).

Networks of hedge funds, communicating with each other and often coordinated explicitly and implicitly into speculative activities can lead to market inefficiencies, thus contradicting a basic hypothesis in finance which assumes that agents are price-takers. In networks, information exchange provides a potential for information asymmetries or at least delays in information [47.145, 146]. In this sense, the existence of networks in their broadest and weakest form may also be a symptom of market breakdown. Analysis of competition in the presence of moral hazard and adverse selection emphasizes the substantial differences between trading of contracts and of contingent commodities. The profit associated with the sale of one unit of a (contingent) good depends then only on its price. Further, the profitability of the sale of one contract may also depend on the identity of the buyer. Identity matters either because the buyer has bought other contracts (the exclusivity problem) or because profitability of the sales depends on the buyer's characteristics, which is also known as the screening problem. Do these issues relate to financial intermediation? Probably yes. Thus financial markets theory has to give a key role to informational and power asymmetries to better understand prices and how they differ from the social values of commodities.

In the presence of proportional transaction costs, no perfect replication strategy is in general available. It is necessary then to define other pricing criteria. Some explicit solutions to the multivariate super-replication problem under proportional transaction costs using a utility maximization problem have been suggested, however. The implication of these and related studies are that super-replication prices are highly expensive and are not acceptable for practical purposes.

Quantitative modeling provides also important sources of incompleteness and at the same time seeks to represent such incompleteness. Research in modeling uncertainty and studies that seek to characterize

mathematically randomness, which can be also sources of incompleteness. We use a number of quantitative approaches including: Brownian motion; long-run memory models and chaos-related approaches ([47.147]; heavy (fat) tails (stable) distributions that, unlike the normal distribution, have very large or infinite variance. These approaches underpin some confusion regarding the definition of uncertainty and how it can be structured in a theory of economics and finance. "G-D does not play dice" (Einstein), "Probabilities do not exist" (Bruno de Finetti) etc. are statements that may put some doubt on the commonly used random-walk hypothesis which underlies martingales finance and markets' efficiency (for additional discussion see [47.148, 149]). These topics are both important and provide open-ended avenues for further and empirical research. In particular, issues of long-run memory and chaos (inducing both very large variances and skewness) are important sources of incompleteness that have been studied intensively [47.150–157]. To deal formally with these issues, *Mandelbrot* and co-workers have introduced both a methodology based on fractal stochastic processes and application of Hurst's 1951 R/S (a range to starndard deviation statistic) methodology (for example, see [47.158–168]). Applications to finance include [47.169–178]). A theoretical extension based on the range process and R/S analysis based on the inverse range process can be found in [47.179, 180] as well as [47.181–183] and [47.184]. Finally, continuous stochastic processes to which risk-neutral pricing can be applied may become incomplete when they are discretized for numerical analysis purposes. Below we shall consider a number of pricing problems in incomplete markets to highlight some of the approaches to asset pricing.

47.6.1 Risk and the Valuation of a Rated Bond

Bonds are not always risk-free. Corporations emitting bonds may default, governments can also default in the payment of their debts, etc. For this reason, rating agencies sell their services and rate firms to assure buyers of the risks they assume when buying the bond. For this reason, the pricing of rated bonds is an important aspect of asset pricing. Below we shall show how such bonds may be valued (see also the earlier bond section). Consider first a non-default coupon-bearing rated bond with a payment of one dollar at maturity T. Further, define the bond m-ratings matrix by a Markov chain $[p_{ij}]$ where $0 \le p_{ij} \le 1$, $\sum_{j=1}^{m} p_{ij} = 1$ denotes the probability that a bond rated i in a given year will be rated j the following one. Discount factors are a function of the rating states, thus a bond rated i has a spot yield R_{it}, $R_{it} \le R_{jt}$ for $i < j$ at time t. As a result, a bond rated i at time t and paying a coupon c_{it} at this time has, as we saw earlier, a value given by

$$B_i(t,T) = c_{it} + \sum_{j=1}^{m} \frac{p_{ij}}{1+R_{jt}} B_j(t+1,T)\,;$$
$$B_{i,T} = \ell_i\,, \quad i = 1,2,3,\ldots m\,,$$

where ℓ_i is the nominal value of a bond rated i at maturity. Usually, $\ell_i = 1$, $i = 1,2,\ldots m-1$, and $\ell_m = 0$ where m is the default state, and there is no recovery in case of default. In vector notation, we have

$$\boldsymbol{B_t} = \boldsymbol{c_t} + \boldsymbol{F_t B_{t+1}}; \quad \boldsymbol{B_T} = \boldsymbol{L}\,,$$

where the matrix $\boldsymbol{F_t}$ has entries $[p_{ij}/(1+R_{jt})]$ and $\boldsymbol{L}$ is a diagonal matrix of entries ℓ_i, $i = 1,2,\ldots,m$. For a zero-coupon bond, we have $\boldsymbol{B}_t = \prod_{k=t}^{T} \boldsymbol{F_k}$. By the same token, rated bonds discounts $q_{it} = 1/(1+R_{it})$ are found by solving the matrix equation

$$\begin{pmatrix} q_{1t} \\ q_{2t} \\ \cdots \\ \cdots \\ q_{mt} \end{pmatrix} = \begin{pmatrix} p_{11}B_{1,t+1} & p_{12}B_{2,t+1} & \cdots\ \cdots & p_{1m}B_{m,t+1} \\ p_{21}B_{1,t+1} & p_{22}B_{2,t+1} & & p_{2m}B_{m,t+1} \\ \cdots & & & \\ \cdots & & & \\ p_{m1}B_{1,t+1} & p_{m2}B_{2,t+1} & & p_{mm}B_{m,t+1} \end{pmatrix}^{-1} \times \begin{pmatrix} B_{1,t} - c_{1t} \\ B_{2,t} - c_{2t} \\ \cdots \\ \cdots \\ B_{m,t} - C_{mt} \end{pmatrix},$$

where at maturity T, $B_i(T,T) = \ell_i$. Thus, in matrix notation, we have: $\bar{\boldsymbol{q}}_t = \boldsymbol{\Gamma}_{t+1}^{-1}(\boldsymbol{B_t} - \boldsymbol{c_t})$. Note that, one period prior to maturity, we have: $\bar{\boldsymbol{q}}_{T-1} = \boldsymbol{\Gamma}_T^{-1}(\boldsymbol{B}_{T-1} - \boldsymbol{c}_{T-1})$, where $\boldsymbol{\Gamma}_T$ is a matrix with entries $p_{ij}B_j(T,T) = p_{ij}\ell_j$.

In order to price the rated bond, consider a portfolio of rated bonds consisting of N_i, $i = 1,2,3\ldots,m$ bonds rated i, each providing ℓ_i dollars at maturity. Let the portfolio value at maturity be equal one dollar. Namely,

$$\sum_{i=1}^{m} N_i \ell_i = 1\,.$$

One period (year) prior to maturity, such a portfolio would be worth $\sum_{i=1}^{m} N_i B_i(T-1,T)$ dollars. By the same token, if we denote by $R_{\mathrm{f},T-1}$ the risk-free discount rate for one year, then assuming no arbitrage, one period

prior to maturity, we have:

$$\sum_{i=1}^{m} N_i B_i(T-1,T) = \frac{1}{1+R_{\mathrm{f},T-1}};$$
$$B_i(T-1,T) = c_{it} + \sum_{j=1}^{m} q_{jt} p_{ij} B_j(T,T);$$
$$B_i(T,T) = \ell_i, \quad i = 1, 2 \ldots, m$$

with $q_{jt} = 1/(1+R_{j,t})$ and $R_{j,t}$ is the one-period discount rate applied to a j rated bond. Assuming no arbitrage, such a system of equations will hold for any of the bonds periods and therefore, we can write the following no-arbitrage condition

$$\sum_{i=1}^{m} N_i B_i(T-k,T) = \frac{1}{\left(1+R_{\mathrm{f},T-k}\right)^k}$$
$$k = 0, 12, 3, \ldots, T,$$

where $R_{\mathrm{f},T-k}$, $k = 1, 2, 3, \ldots$, is the risk-free rate term structure which provides a system of $T+1$ equations spanning the bond life. In matrix notation this is given by

$$\boldsymbol{N}\boldsymbol{B}_{\boldsymbol{T}-k} = \frac{1}{\left(1+R_{\mathrm{f},T-k}\right)^k},$$
$$k = 0, 1, 2, \ldots, T; \quad \boldsymbol{N} = (N_1, N_2, \ldots, N_m);$$
$$\boldsymbol{B}_{\boldsymbol{T}-k} = (B_{1,T-k}, B_{2,T-k}, \ldots, B_{m,T-k}).$$

As a result, assuming that the bond maturity is larger than the number of ratings ($T \geq m+1$), the hedging portfolio of rated bonds is found by a solution of the system of linear equations above, leading to the unique solution:

$$\boldsymbol{N}^* = \Im^{-1}\boldsymbol{\Omega},$$

where $\Im$ is the matrix transpose of $[B_{i,T-j+1}]$ and $\boldsymbol{\Omega}$ is a column vector with entries $\left[1/(1+R_{\mathrm{f},T-s})^s\right]$, $s = 0, 1, 2 \ldots m-1$. Explicitly, we have:

$$\begin{pmatrix} N_1 \\ N_2 \\ \ldots \\ \ldots \\ N_m \end{pmatrix} = \begin{pmatrix} B_{1,T} & B_{2,T} & B_{3,T} & \ldots & B_{m,T} \\ B_{1,T-1} & B_{2,T-1} & B_{3,T-1} & \ldots & B_{m,T-1} \\ \ldots & & & & \ldots \\ \ldots & & & & \ldots \\ B_{1,T-m} & B_{2,T-m} & B_{3,T-m} & \ldots & B_{m,T-m} \end{pmatrix}^{-1} \times \begin{pmatrix} 1 \\ 1/(1+R_{\mathrm{f},T-1})^1 \\ \ldots \\ \ldots \\ 1/(1+R_{\mathrm{f},T-m})^m \end{pmatrix}.$$

Thus, a condition for no arbitrage is given by the system of nonlinear equations

$$\Im^{-1}\boldsymbol{\Omega}\boldsymbol{B}_{\boldsymbol{T}-k} = \frac{1}{\left(1+R_{\mathrm{f},T-k}\right)^k}$$
$$k = m, m+1, \ldots, T.$$

For example, for a zero-coupon rated bond and stationary short discounts, we have $\boldsymbol{B}_{t-k} = (\boldsymbol{F})^k$ and therefore, the no-arbitrage condition becomes:

$$\Im^{-1}\boldsymbol{\Omega}\boldsymbol{F}^k = \frac{1}{\left(1+R_{\mathrm{f},T-k}\right)^k} \quad k = m, m+1, \ldots, T$$

where $\boldsymbol{F}$ has entries $q_j p_{ij}$. This system of equations therefore provides $T+1-m$ equations applied to determining the bond ratings short (one-period) discount rates q_j. Our system of equations may be over- or under-identified for determining the ratings discount rates under our no-arbitrage condition, however. Of course, if $T+1-m = m$, we have exactly m additional equations we can use to solving the discount rates uniquely (albeit, these are nonlinear equations and can only be solved numerically).

Otherwise, the rated bond market is incomplete and we must proceed to some approach that can, nevertheless, provide an estimate of the discount rates. We use for convenience a sum of squared deviations from the rated bond arbitrage condition, in which case we minimize the following expression:

$$\min_{0 \leq q_1, q_2, \ldots q_{m-1}, q_m \leq 1} \sum_{k=m}^{T} \left(\Im^{-1}\boldsymbol{\Omega}\boldsymbol{B}_{\boldsymbol{T}-k} - \frac{1}{\left(1+R_{\mathrm{f},T-k}\right)^k} \right)^2.$$

Further additional constraints, reflecting expected and economic rationales of the ratings discounts q_j, might be added, such as:

$$0 \leq q_j \leq 1 \quad \text{and}$$
$$0 \leq q_m \leq q_{m-1} \leq q_{m-2} \leq q_{m-3}, \ldots \leq q_2 \leq q_1 \leq 1.$$

These are typically nonlinear optimization problems however. A simple two-rating example highlights some of the complexities in determining both the hedging portfolio and the ratings discounts provided the risk-free term structure is given. When the bond can default, we have to proceed as shown below.

47.6.2 Valuation of Default-Prone Rated Bonds

Let the first time n, a bond rated initially i, is rated j and let the probability of such an event be $f_{ij}(n)$. This probability equals the probability of not having gone through a j-th rating in prior transitions and be rated j at time n. For transition in one period, this is equal the transition bond rating matrix (S&P or Moody's matrix, as stated earlier), while for a transition in two periods it equals the probability of transition in two periods conditional on not having reached rating j in the first period. In other words, we have:

$$f_{ij}(1) = p_{ij}(1) = p_{ij}; \quad f_{ij}(2) = p_{ij}(2) - f_{ij}(1)p_{jj}$$

and generally, by recursion,

$$f_{ij}(n) = p_{ij}(n) - \sum_{k=1}^{n-1} f_{ij}(k)p_{jj}(n-k)\,.$$

The probability of a bond defaulting (and not defaulting) prior to time n is thus,

$$F_{km}(n-1) = \sum_{j=1}^{n-1} f_{km}(j)\,;$$
$$\bar{F}_{km}(n-1) = 1 - F_{km}(n-1)\,.$$

At present, denote by $\Phi_i(n)$ the probability that the bond is rated i at time n. In vector notation we write $\bar{\Phi}(n)$. Thus given the rating matrix $[\boldsymbol{P}]$, we have:

$$\bar{\Phi}(n) = [\boldsymbol{P}]^{'}\bar{\Phi}(n-1)\,,$$
$$n = 1, 2, 3, \ldots \quad \text{and} \quad \bar{q}(0) \text{ given}\,,$$

where $[\boldsymbol{P}]^{'}$ is the matrix transpose. Thus, at n, $\bar{\Phi}(n) = [\boldsymbol{P}^{'}]^n\bar{\Phi}(0)$. The present value of a coupon payment at time n (given that there was no default at this time) is therefore discounted at the yield $R_{j,n}$, $q_{j,n} = 1/(1+R_{j,n})$ if the bond is rated j. In other words, its present value is

$$\sum_{j=1}^{m-1} c_{j,n}q_{j,n}^n\Phi_{j,n}; \quad \Phi_{j,n} = \sum_{i=1}^{m-1}\Phi_{i,0}p_{ij}^{(n)}\,,$$

where $p_{ij}^{(n)}$ is the ij-th entry of the transposed power matrix $[\boldsymbol{P}^{'}]^n$ and $\Phi_{i,0}$ is the probability that initially the bond is rated i.

When a coupon-bearing default bond rated i at time s defaults at time $s+1$, $T-(s+1)$ periods before maturity with probability $f_{im}(s+1-s)$, we have a value:

$$V_{s,i} = \left(c_{i,T-s} + q_i\ell_{m,T-(s+1)}\right) \text{ w. p. } f_{im}(1)\,.$$

If such an event occurs at time $s+2$, with probability $f_{im}(s+2-s) = f_{im}(2)$, we have:

$$V_{s,i} = \left(c_{i,T-s} + \sum_{k=1}^{m-1} q_k c_{k,T-(s+1)}\Phi_{k,(s+1)-s} + q_i^2\ell_{m,T-(s+2)}\right) \text{ w. p. } f_{im}(2)\,,$$

where $\Phi_{k,1} = \sum_{i=1}^{m-1}\Phi_{i,0}p_{ik}^{(1)}$ and $\Phi_{i,0}$ is a vector whose entries are all zero except at i (since at s we conditioned the bond value to a rating i). By the same token three periods hence and prior to maturity, we have

$$V_{s,i} = \left(c_{i,T-s} + \sum_{k=1}^{m-1} q_k c_{k,T-(s+1)}\Phi_{i,0}p_{ik}^{(1)} + \sum_{k=1}^{m-1} q_k^2 c_{k,T-(s+2)}\Phi_{i,0}p_{ik}^{(2)} + q_i^3\ell_{m,T-(s+3)}\right) \text{ w. p. } f_{im}(3)\,.$$

And generally, for any period prior to maturity,

$$V_{s,i} = \left(c_{i,T-s} + \sum_{\theta=1}^{\tau-1}\sum_{k=1}^{m-1} q_k^{\theta} c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)} + q_i^{\tau}\ell_{m,T-(s+\tau)}\right) \text{ w. p. } f_{im}(\tau)\,.$$

In expectation, if the bond defaults prior to its maturity, its expected price at time s,

$$EB_{i,\mathrm{D}}(s,T) = c_{i,T-s} + \sum_{\tau=1}^{T-s}\left(q_i^{\tau}\ell_{m,T-(s+\tau)} + \sum_{\theta=1}^{\tau-1}\sum_{k=1}^{m-1} q_k^{\theta} c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)}\right) \times f_{im}(\tau)\,,$$

where $\ell_{m,T-j}$ is a portion of the bond nominal value that the bondholder recuperates when the bond defaults and which is assumed to be a function of the time remaining for the bond to be redeemed. And therefore, the price of

such a bond is:

$$B_{i,\mathrm{ND}}(s,T)=\left(c_{i,T-s}+\sum_{k=1}^{m-1}q_k^{T-s}\ell_k\Phi_{i,0}p_{ik}^{(T-s)}+\sum_{\theta=1}^{T-s-1}\sum_{k=1}^{m-1}q_k^{\theta}c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)}\right)\times\left[1-\sum_{u=1}^{T-s}f_{im}(u)\right]$$

where ℓ_i denotes the bond nominal value at redemption when it is rated i. Combining these sums, we obtain the price of a default-prone bond rated i at time s

$$\begin{aligned}B_i(s,T)=c_{i,T-s}+&\left(c_{i,T-s}+\sum_{k=1}^{m-1}q_k^{T-s}\ell_k\Phi_{i,0}p_{ik}^{(T-s)}+\sum_{\theta=1}^{T-s-1}\sum_{k=1}^{m-1}q_k^{\theta}c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)}\right)\times\left[1-\sum_{u=1}^{T-s}f_{im}(u)\right]\\&+\sum_{\tau=1}^{T-s}\left(q_i^{\tau}\ell_{m,T-(s+\tau)}+\sum_{\theta=1}^{\tau-1}\sum_{k=1}^{m-1}q_k^{\theta}c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)}\right)f_{im}(\tau)\,.\end{aligned}$$

Finally, for a zero-coupon bond, this is reduced to

$$B_i(s,T)=c_{i,T-s}+\left(\sum_{k=1}^{m-1}q_k^{T-s}\ell_k\Phi_{i,0}p_{ik}^{(T-s)}\right)\times\left[1-\sum_{u=1}^{T-s}f_{im}(u)\right]+\sum_{\tau=1}^{T-s}\left(q_i^{\tau}\ell_{m,T-(s+\tau)}\right)f_{im}(\tau)\,.$$

To determine the price (discounts rates) for a default-prone rated bond we can proceed as before by constructing a hedging portfolio consisting of $N_1, N_2, \ldots, N_{m-1}$ shares of bonds rated $i=1,2,\ldots,m-1$. Again, let $R_{\mathrm{f},T-u}$ be the risk-free rate when there are u periods left to maturity. Then, assuming no arbitrage and given the term structure risk-free rate, we have:

$$\sum_{i=1}^{m-1}N_iB_i(s,T)=\frac{1}{\left(1+R_{\mathrm{f},T-s}\right)^s},\quad s=0,1,2,\ldots$$

with $B_i(s,T)$ defined above. Note that the portfolio consists of only $m-1$ rated bonds and therefore, we have in fact $2m-1$ variables to be determined based on the risk-free term structure. Assuming that our system is over- (or under-)determined, we are reduced to solving the following minimum squared deviations problem:

$$\min_{\substack{0\le q_1\le q_2\le\ldots\le q_{m-1}\le 1;\\ N_1,N_2,N_3\ldots,N_{m-1}}}\sum_{s=0}^{T}\left[\sum_{k=1}^{m-1}N_kB_k(s,T)-\frac{1}{\left(1+R_{\mathrm{f},T-s}\right)^s}\right]^2$$

subject to:

$$\begin{aligned}B_i(s,T)=c_{i,T-s}+&\left(c_{i,T-s}+\sum_{k=1}^{m-1}q_k^{T-s}\ell_k\Phi_{i,0}p_{ik}^{(T-s)}+\sum_{\theta=1}^{T-s-1}\sum_{k=1}^{m-1}q_k^{\theta}c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)}\right)\times\left[1-\sum_{u=1}^{T-s}f_{im}(u)\right]+\sum_{\tau=1}^{T-s}\left(q_i^{\tau}\ell_{m,T-(s+\tau)}\right.\\&\left.+\sum_{\theta=1}^{\tau-1}\sum_{k=1}^{m-1}q_k^{\theta}c_{k,T-(s+\theta)}\Phi_{i,0}p_{ik}^{(\theta)}\right)f_{im}(\tau)\,.\end{aligned}$$

This is of course a nonlinear optimization problem which can be solved analytically with respect to the hedged portfolio, and use the remaining equations to calculate the ratings discount rates. A solution can be found numerically. Such an analysis is a straightforward exercise however. Below we consider some examples.

Example 47.5 (a two-rated default bond): Consider a two-rated zero-coupon bond and define the transition matrix

$$\boldsymbol{P}=\begin{pmatrix}p & 1-p\\ 0 & 1\end{pmatrix}\quad\text{with}\quad\boldsymbol{P}^n=\begin{pmatrix}p^n & 1-p^n\\ 0 & 1\end{pmatrix}.$$

The probability of being in one of the two states after n periods is $(p^n, 1-p^n)$. Further,

$$\begin{aligned}f_{12}(1)&=1-p\,;\\ f_{12}(2)&=p_2^{(2)}-(1)f_{12}(1)=1-p^2-(1-p)\\&=p(1-p)\,.\end{aligned}$$

Thus, for a non-coupon-paying bond, we have:

$$B_i(s,T) = c_{i,T-s} + q^{T-s}\ell\left[1 - \sum_{u=1}^{T-s} f_{12}(u)\right] + \sum_{\tau=1}^{T-s}\left(q^{\tau}\ell_{m,T-(s+\tau)}\right) f_{12}(\tau)\,.$$

In particular,

$$\begin{aligned}
B_1(T,T) &= \ell\,,\\
B_1(T-1,T) &= q\ell\,[1-f_{12}(1)] + q\ell_{m,0} f_{12}(1)\,,\\
B_1(T-2,T) &= q^2\ell\,[1-f_{12}(1)-f_{12}(2)]\\
&\quad + q\ell_{m,1} f_{12}(\tau) + q^2\ell_{m,0} f_{12}(2)\,,\\
B_1(T-3,T) &= q^3\ell\,[1-f_{12}(1)-f_{12}(2)-f_{12}(3)]\\
&\quad + \left(q\ell_{m,2}\right) f_{12}(1) + q^2\ell_{m,1} f_{12}(2)\\
&\quad + q^3\ell_{m,0} f_{12}(3)\,.
\end{aligned}$$

If we have a two-year bond, then the condition for no arbitrage is

$$\begin{aligned}
NB_1(T,T) &= N\ell = 1 \quad \text{and} \quad N = 1/\ell\,,\\
NB_1(T-1,T) &= \frac{1}{1+R_{\mathrm{f},T-1}} \Rightarrow 1 + R_{1,T-1}\\
&= \frac{1+R_{\mathrm{f},T-1}}{1-\left(1-\ell_{m,0}/\ell\right) f_{12}(1)}\,.
\end{aligned}$$

If we have a two-year bond, then the least quadratic deviation cost rating can be applied. Namely,

$$\min_{0\le q\le 1} \mathbb{Q} = \left[(1/\ell)B\,(T-1,T) - \left(q_{f,T-1}\right)\right]^2 + \left[(1/\ell)B\,(T-2,T) - \left(q_{f,T-2}\right)\right]^2\,.$$

Subject to:

$$\begin{aligned}
B_1(T-1,T) &= q\ell\,[1-f_{12}(1)] + q\ell_{m,0} f_{12}(1),\\
B_1(T-2,T) &= q^2\ell\,[1-f_{12}(1)-f_{12}(2)]\\
&\quad + q\ell_{m,1} f_{12}(\tau) + q^2\ell_{m,0} f_{12}(2)\,.
\end{aligned}$$

Leading to a cubic equation in q that we can solve by the usual methods. Rewriting the quadratic deviation in terms of the discount rate yields:

$$\min_{0\le q\le 1} \left(q\left\{1-f_{12}(1)\left[1-(\ell_{m,0}/\ell)\right]\right\} - \left(q_{f,T-1}\right)\right)^2 + \left\{q^2\left[1-f_{12}(1)-(1-\ell_{m,0}/\ell) f_{12}(2)\right] + q(\ell_{m,1}/\ell) f_{12}(1) - \left(q_{f,T-2}\right)\right\}^2\,.$$

Set

$$\begin{aligned}
a &= \left\{1-f_{12}(1)\left[1-(\ell_{m,0}/\ell)\right]\right\}\,;\\
b &= \left[1-f_{12}(1)-(1-\ell_{m,0}/\ell) f_{12}(2)\right]\,;\\
c &= (\ell_{m,1}/\ell) f_{12}(1)\,.
\end{aligned}$$

Then an optimal q is found by solving the equation

$$2q^3b^2 + 3q^2bc + q\left(a^2 - 2bq_{f,T-2} + c^2\right) - \left(aq_{f,T-1} + cq_{f,T-2}\right) = 0\,.$$

Assume the following parameters,

$$\begin{aligned}
&R_{\mathrm{f},T-1} = 0.07; \quad R_{\mathrm{f},T-1} = 0.08, \quad p = 0.8,\\
&\ell = 1, \quad \ell_{m,0} = 0.6, \quad \ell_{m,1} = 0.4\,.
\end{aligned}$$

In this case, $f_{12}(1) = 1-p = 0.2$ and $f_{12}(2) = p(1-p) = 0.16$. For a one-period bond, we have

$$1 + R_{1,T-1} = \frac{1+0.07}{1-(0.084)} = 1.168$$

and therefore we have a 16.8% discount, $R_{1,T-1} = 0.168$. For a two-period bond, we have instead (using the minimization technique): $a = 0.92$, $b = 0.736$, $c = 0.084$ and therefore,

$$q^3 + 0.171\,129\,q^2 - 0.470\,28\,q - 0.865\,33 = 0\,.$$

Whose solution provides q and therefore $1 + R_{1,T-1}$.

47.6.3 "Engineered" Risk-Neutral Distributions and Risk-Neutral Pricing

When a market is complete, an asset price can be defined as follows:

$$S_t = \mathrm{e}^{-R_{\mathrm{f}}(T-t)} E_t^*\left(S_T\,|\Omega_t\right)\,,$$

where Ω_t is a filtration, meaning that the expectation is calculated on the basis of all the information available up to time t and the probability distribution with respect to which the expectation is taken is a risk-neutral distribution. That is:

$$E_t^*\left(S_T\,|\Omega_t\right) = \int S_T\,\mathrm{d}F_{T|t}\,, \quad T > t$$

where $F_{T|t}$ is the asset risk-neutral distribution at time T based on the data available at time t. If the underlying price process is given by a stochastic process, then $E_t^*\left(S_T\,|\Omega_t\right)$ is the optimal forecast (filter) estimate of the asset price using the distribution $F_{T|t}$. In such circumstances, and if such a distribution exists, then derived assets such as call and put options are also priced by

$$C_t = \mathrm{e}^{-R_{\mathrm{f}}(T-t)} E_t^*\left(C_T\,|\Omega_t\right) = \int_K C_T\,\mathrm{d}F_{T|t}\,\left(S_T\right)\,,$$

$$C_T = \max(S_T - K, 0), \quad T > t\,,$$

$$P_t = \mathrm{e}^{-R_\mathrm{f}(T-t)} E_t^* \left(P_T \,|\Omega_t\right) = \int_0^K P_T \,\mathrm{d}F_{T|t}\,(S_T)\ ,$$

$$P_T = \max(K - S_T, 0), \quad T > t\,.$$

Of course, if the distribution happens to be normal then the assets prices equal the discounted best mean forecast of the future asset price. When this is not the case and markets are incomplete, asset pricing in practice seeks to determine the risk-neutral distribution that allows application of risk-neutral pricing whether markets are in fact complete or incomplete (for an empirical study see [47.185, 186], for example). There are numerous sources of information, and approaches used to *engineer* such a distribution. For example, let there be m derived assets x_t^j and let there be some data points up to time t regarding these assets. In this case, for each time t, the optimal least-square estimate of the risk-neutral distribution is found by minimizing the least squares below by a selection of the appropriate parameters defining the underlying price process:

$$\sum_{i=0}^{t} \left\{ \sum_{j=1}^{m} \left[x_i^j - \mathrm{e}^{-R_\mathrm{f}(T-i)} E_i^* \left(x_T^j \,|\Omega_i \right) \right]^2 + \left[S_i - \mathrm{e}^{-R_\mathrm{f}(T-i)} E_i^* \left(S_T \,|\Omega_i\right) \right]^{\ddot{}2} \right\} .$$

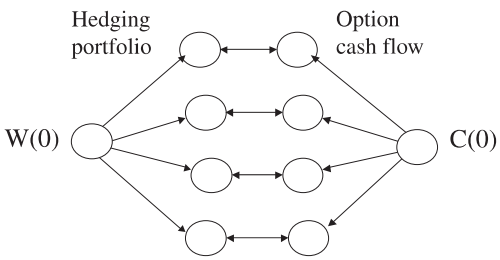
Here x_i^j is an actual observation of asset j taken at time i. If prices are available over several specific time periods (for example, an option for three months, six months and a year), then summing over available time periods we will have:

$$\sum_{i=0}^{t} \left\{ \sum_{\ell=1}^{L} \sum_{j=1}^{m} \left[x_i^j - \mathrm{e}^{-R_\mathrm{f}(T_\ell - i)} E_i^* \left(x_{T_\ell}^j \mid \Omega_i \right) \right]^2 + \sum_{k=i+1} \left[S_i - \mathrm{e}^{-R_\mathrm{f}(k-i)} E_i^* \left(S_k \mid \Omega_i\right) \right]^{\ddot{}2} \right\} .$$

Of course, other techniques can be taken in this spirit, providing thereby the optimal distribution forecast estimate. This is a problem that is over-parameterized however and therefore some assumptions are often made to reduce the number of parameters that define the presumed risk-neutral distribution. Examples and applications are numerous. Some authors assume a general multi-parameter distribution (such as the Burr distribution) for the risk-neutral distribution and calculate the parameters. Others seek the distribution outright while others assume an underlying process and calculate the best fit parameters. Both discrete-time and continuous-time models are used. Other models assume a broader framework such as a stochastic process with or without stochastic volatility with parameters to be estimated based on data availability. Below we shall consider a number of such cases (see also [47.187, 188].

Example 47.6 (mean variance replication hedging): This example consists of constructing a hedging portfolio in an incomplete (stochastic volatility) market by equating *as much as possible* cash flows resulting from a hedging portfolio and option prices. We shall do so while respecting the basic rules of rational expectations and risk-neutral pricing. This implies that at all times the price of the portfolio and the option price are the same. Let $W(t)$ be the portfolio price and $C(t)$ be the option price. At time $t = 0$, we evidently have as well $W(0) = C(0)$, similarly at some future date.

$$\begin{array}{ccc} W(0) & \Leftarrow & \tilde{W}(1) \\ \Downarrow & & \Updownarrow \\ C(0) & & \tilde{C}(1) \end{array}$$



$$\tilde{W}(1) = \tilde{C}(1)$$

However, under risk-neutral pricing we have:

$$C(0) = \frac{1}{1+R_\mathrm{f}} E\tilde{C}(1); \quad W(0) = \frac{1}{1+R_\mathrm{f}} E\tilde{W}(1)$$

$$\text{and} \quad C(0) = W(0) \quad \text{and} \quad E\tilde{W}^2(1) = \tilde{E}C^2(1)\,.$$

These provide three equations only. Since a hedging portfolio can involve a far greater number of parameters, it might be necessary to select an objective to minimize. A number of possibilities are available.

Rubinstein [47.189], as well as *Jackwerth* and *Rubinstein* [47.190] for example, suggested a simple quadratic

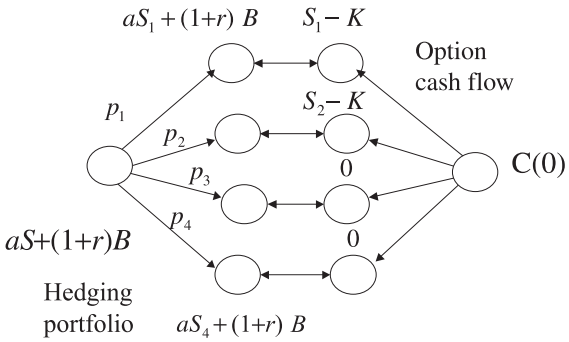
optimization problem by minimizing the quadratic difference of the probabilities associated to the binomial tree. Alternatively, a quadratic objective that leads to the minimization of a hedging portfolio and the option ex-post values of some option contract with risk-neutral pricing leads to:

$$\min_{p_1,\dots p_n} \Phi = E\left(\tilde{W}(1)-\tilde{C}(1)\right)^2 . \quad \text{Subject to:}$$

$$W(0)=C(0) \quad \text{or}$$

$$\frac{1}{1+R_\mathrm{f}}\left[E\tilde{W}(1)\right]=\frac{1}{1+R_\mathrm{f}}\left[E\tilde{C}(1)\right] \quad \text{and}$$

$$E\tilde{W}^2(1)=\tilde{E}C^2(1)$$



Of course the minimizing objective can be simplified further to:

$$\min_{p_1,\dots p_n} \Phi = E\tilde{C}^2(1)-E\tilde{W}(1)\tilde{C}(1) \quad \text{or}$$

$$\min_{p_1,\dots p_n} \Phi = \sum_{i=1}^{n} p_i\left(C_{1i}^2-W_{1i}C_{1i}\right),$$

where W_{1i}, C_{1i} are the hedging portfolio and option outcomes associated with each of the events i, which occur with probability p_i, $i=1,2,\dots n$.

Example 47.7: Define by S_j, $j=1,2,\dots n$ the n states a stock can assume at the time an option can be exercised. We set, $S_0<S_1<S_2<\dots<S_n$ and define the buying and selling prices of the stock by S^a, S^b, respectively. By the same token, define the corresponding observed call option prices C^a, C^b. Let p be the probability of a price increase. Of course, if the ex-post price is S_n, this will correspond to the stock increasing each time period with probability $P_n=(n,n)^\top p^n(1-p)^{n-n}=p^n$. By the same token, the probability of the stock having a price S_j corresponding to the stock increasing j times and decreasing $n-j$ times is given by the binomial probability

$$P_j=\binom{n}{j}p^j(1-p)^{n-j} .$$

As a result, we have under risk-neutral pricing:

$$S=\frac{1}{1+R_\mathrm{f}}\sum_{j=0}^{n}P_jS_j$$

$$=\frac{1}{1+R_\mathrm{f}}\sum_{j=0}^{n}S_j\binom{n}{j}p^j(1-p)^{n-j};$$

$$S^a\le S\le S^b$$

while the call option price is

$$C=\sum_{j=0}^{n}C_j$$

$$=\frac{1}{(1+R_\mathrm{f})^n}\sum_{j=0}^{n}\binom{n}{j}p^j(1-p)^{n-j}$$

$$\times\max(S_j-K,0)$$

with an appropriate constraint on the call option value $C^a\le C\le C^b$. Note that S, C as well as p are the only unknown values so far. While the buy and sell values for stock and options, the strike time n and its price K as well as the discount rate and future prices S_j are given. Our problem at present is to select an objective which will make it possible to obtain risk-neutral probabilities. We can do so by minimizing the quadratic distance between a portfolio of a unit of stock and a bond B. At n, the portfolio is equal $aS_j+(1+R_\mathrm{f})^nB$ if the price is S_j. Of course, initially, the portfolio equals:

$$S=\frac{1}{(1+R_\mathrm{f})^n}\left[a\sum_{j=0}^{n}P_jS_j+(1+R_\mathrm{f})^nB\right] .$$

As a result, the least-squared replicating portfolio is given by:

$$\Phi=\sum_{j=1}^{n}P_j\left[aS_j+(1+R_\mathrm{f})^nB-\max(S_j-K,0)\right]^2$$

which leads to the following optimization problem

$$\min_{1\geq p\geq 0, C, S} \Phi = \sum_{j=1}^{n} \binom{n}{j} p^j (1-p)^{n-j} \times \left[aS_j + (1+R_f)^n B - \max(S_j - K, 0)\right]^2 .$$

Subject to:

$$S = \frac{1}{(1+R_f)^n} \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} S_j ;$$

$$S^a \leq S \leq S^b,$$

$$C = \frac{1}{(1+R_f)^n} \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} \times \max(S_j - K, 0); \quad C^a \leq C \leq C^b ,$$

$$aS + B = C .$$

The numerical solution of this problem is straightforward.

Example 47.8 (fitting continuous risk-neutral distributions): The simplest such model of course consists of using an underlying binomial stock process and using options data (call and put) to estimate the stock process parameters and risk-neutral distribution. Assume that we assume a theoretical mixture price model given by:

$$\frac{dS}{S} = \begin{cases} \alpha_j \, dt + \sigma_j \, dW_j & \pi_j \\ j = 1, 2, \ldots, m & \\ \pi_j \geq 0 & \sum_{j=1}^{m} \pi_j \end{cases} , \quad S(0) = S_0 .$$

The solution of this model in terms of a risk-neutral numeraire is, (as we saw earlier):

$$S(t) = S(0) \sum_{j=1}^{m} \pi_j \, e^{\left(R_f - \frac{\sigma_j^2}{2}\right)t + \sigma_j W_t^*(t)} ,$$

$$W_t^*(t) = W_j(t) + \frac{\alpha_j - R_f}{\sigma_j} t .$$

Let there be call and put options prices given by:

$$C_t = e^{-R_f(T-t)} E_t^* (C_T \,|\Omega_t) ,$$
$$C_T = \max(S_T - K, 0), \quad T > t ,$$
$$P_t = e^{-R_f(T-t)} E_t^* (P_T \,|\Omega_t) ,$$
$$P_T = \max(K - S_T, 0), \quad T > t .$$

A simple least-squares optimization problem that seeks to calculate the underlying process parameters is then:

$$\min_{\alpha_j, \sigma_j, \pi_j} \sum_{i=0}^{t} \left\{ \left[C_i - e^{-R_f(T-i)} E_i^* (C_T \,|\Omega_i) \right]^2 + \left[P_i - e^{-R_f(T-i)} E_i^* (P_T \,|\Omega_i) \right]^2 \right\} ,$$

where

$$C_t = e^{-R_f(T-t)} E_t \left\{ \max \left[S(0) \times \sum_{j=1}^{m} \pi_j \, e^{\left(R_f - \frac{\sigma_j^2}{2}\right)t + \sigma_j W^*(t)} - K, 0 \right] |\Omega_t \right\} ,$$

$$P_t = e^{-R_f(T-t)} E_t \left\{ \max \left[K - S(0) \times \sum_{j=1}^{m} \pi_j \, e^{\left(R_f - \frac{\sigma_j^2}{2}\right)t + \sigma_j W^*(t)}, 0 \right] |\Omega_t \right\} .$$

Of course such problems can be solved by MATLAB or by nonlinear optimization routines. Further refinements can be developed by noting that the stock price must also meet the risk-neutral condition at each time prior to time t. A simple case is obtained when we consider the mixture of two such log-normal models. Such an assumption will imply of course that prices are skewed. Explicitly, assume a mixture of the log-normal distributions

$$f(S_T) = \begin{cases} L(\alpha_1, \beta_1) & \text{w.p. } \theta \\ L(\alpha_2, \beta_2) & \text{w.p. } 1-\theta \end{cases} ;$$

$$\hat{f}(S_T) \sim \theta L(\alpha_1, \beta_1) + (1-\theta) L(\alpha_2, \beta_2) .$$

For such a process we can use the normal mixtures as follows:

$$P(\tilde{R} \leq R) = \theta N\left(\frac{\log R - \mu_1}{\sigma_1}\right) + (1-\theta) N\left(\frac{\log R - \mu_2}{\sigma_2}\right) .$$

Elementary but tedious analysis of the moments can provide an estimate of the variance, the distribution

skewness and its kurtosis, or

$$\begin{aligned}
\text{var} &= \theta_1\sigma_1^2 + \theta_2\sigma_2^2 + \theta_1\theta_2\,(\mu_1-\mu_2)^2\ , \\
\text{Skewness} &= \frac{\theta_1\theta_2\,(\mu_1-\mu_2)}{(\text{var})^{3/2}} \\
&\quad\times\left[3\left(\sigma_1^2-\sigma_2^2\right)+(\theta_2-\theta_1)\,(\mu_1-\mu_2)^2\right], \\
\text{Kurtosis} &= \frac{3\left(\theta_1\sigma_1^4+\theta_2\sigma_2^4\right)}{(\text{var})^2} \\
&\quad+\frac{6\theta_2\theta_1\,(\mu_1-\mu_2)^2\left(\theta_2\sigma_1^1+\theta_1\sigma_2^2\right)}{(\text{var})^2} \\
&\quad+\frac{\theta_2\theta_1\,(\mu_1-\mu_2)^4\left(\theta_1^3+\theta_2^3\right)}{(\text{var})^2}\ .
\end{aligned}$$

These moments of the mixture process indicate the behavior of the underlying risk-neutral distribution, highlighting the *market* intention as it is reflected in these moments (skew to the right or to the left of the uncertainty regarding future asset prices). To estimate the mixture process parameters we may then fit the data to say, call and put data, freely available on the appropriate market. Let:

$\hat{C}_{i,T}$ = European Call option price at $i = 1, \ldots m$;

$\hat{P}_{i,T}$ = European Put option price at $i = 1, 2, \ldots m$

with

$$\hat{C}(S, K_i, T) = \mathrm{e}^{-R_\mathrm{f}T}\int_{K_i}^{\infty}(S_T - K_i)\hat{f}(S_T)\mathrm{d}S_T\ ,$$

$$\hat{P}(S, K_i, T) = \mathrm{e}^{-R_\mathrm{f}T}\int_{0}^{K_i}(K_i - S_T)\hat{f}(S_T)\mathrm{d}S_T\ .$$

Table 47.1 Comparison of the log-normal and bi-log-normal model

Date	Number of observations	Fit (sum of squares)	
		Log-normal	Bi-log-normal
09/06/2002	24	0.121	0.121
10/06/2002	24	0.060	0.060
17/06/2002	26	0.022	0.011
15/06/2002	23	0.070	0.004
10/07/2002	19	0.038	0.008
12/08/2002	16	0.022	0.005
Average	21.7	0.056	0.038

The resulting (data fit) optimization problem is then

$$\min_{\alpha_1,\beta_1;\alpha_2,\beta_2;\theta}\left[\sum_{i=1}^{m}\left(\hat{C}_{i,T}-C_{i,T}\right)^2 + \sum_{j=1}^{m}\left(\hat{P}_{j,T}-P_{j,T}\right)^2\right].$$

Subject to the call and put theoretical price estimates. In other words, introducing the time-dependent prices for the call and put options, we have

$$\min_{\alpha_1,\beta_1;\alpha_2,\beta_2;\theta}\sum_{t=0}^{T}\left[\sum_{i=1}^{m}\left(\hat{C}_{i,t}-C_{i,t}\right)^2 + \sum_{j=1}^{m}\left(\hat{P}_{j,t}-P_{j,t}\right)^2\right].$$

Subject to:

$$C_{it} = \mathrm{e}^{-R_\mathrm{f}(T-t)}\int_{K_i}^{\infty}(S_T - K_i)\hat{f}_t(S_T)\mathrm{d}S_T \quad \text{and}$$

$$\hat{P}_{it} = \mathrm{e}^{-R_\mathrm{f}(T-t)}\int_{0}^{K_i}(K_i - S_T)\hat{f}_t(S_T)\mathrm{d}S_T\ ,$$

where $\hat{f}_t(S_T)$ is given by the underlying multi-parameter mixture log-normal process. *Stein* and *Hecht* (Bank of Israel, Monetary Devision) use instead the following objective, which they have found more stable using data of the Israeli shekel and the American dollar.

$$\min_{\alpha_1,\beta_1;\alpha_2,\beta_2;\theta}\sum_{t=0}^{T}\left[\sum_{i=1}^{m}\left(1-\frac{\hat{C}_{i,t}}{C_{i,t}}\right)^2 + \sum_{j=1}^{m}\left(1-\frac{\hat{P}_{i,t}}{P_{i,t}}\right)^2\right]$$

Using Israeli and US currency data and comparing a log-normal and a bi-log-normal model they show that the bi-log-normal model provides a better fit, as shown in Table 47.1.

For each of these periods, they calculated as well the parameters of the underlying exchange rate process. As a result, they were able to estimate the probability of currency devaluation and that of appreciation of the shekel versus the dollar. Clearly, the result in Table 47.1 point out to a better fit when the bi-lognormal model is used.

Some authors simplify the computation of implied parameters by considering a multi-parameter distribution. In other words, this approach assumes outright that

the risk-neutral distribution can be approximated by a known distribution. For simplicity, assume first a two-parameter (c, ζ) Weibull distribution given by:

$$f(\tau) = \frac{c}{\zeta}\left(\frac{\tau}{\zeta}\right)^{c-1} \mathrm{e}^{-(\tau/\zeta)^c}, \quad \tau \geq 0, \zeta, \quad c > 0;$$
$$F(\tau) = 1 - \mathrm{e}^{-(\tau/\zeta)^c},$$
$$E(\tau) = \zeta \Gamma\left(\frac{1}{c} + 1\right),$$
$$\mathrm{var}(\tau) = \zeta^2\left[\Gamma\left(\frac{2}{c} + 1\right) - \Gamma^2\left(\frac{1}{c} + 1\right)\right].$$

For the call option, calculations are easily applied and we have after some elementary manipulations:

$$\hat{C}_{it} = \mathrm{e}^{-R_\mathrm{f}(T-t)} \int_{K_i}^{\infty} (S_T - K_i) \hat{f}_t(S_T) \mathrm{d}S_T$$
$$= \mathrm{e}^{-R_\mathrm{f}(T-t)} c \int_{K_i}^{\infty} \left(\frac{S_T}{\zeta}\right)^c \mathrm{e}^{-(S_T/\zeta)^c} \mathrm{d}S_T$$
$$- \mathrm{e}^{-R_\mathrm{f}(T-t)} K_i \mathrm{e}^{-(K_i/\zeta)^c}.$$

If we set:

$$\left(\frac{S_T}{\zeta}\right)^c = u, \quad c\left(\frac{1}{\zeta}\right)\left(\frac{S_T}{\zeta}\right)^{c-1} \mathrm{d}S_T = \mathrm{d}u \quad \text{then},$$
$$\hat{C}_{it} = \mathrm{e}^{-R_\mathrm{f}(T-t)} c \int_{\left(\frac{K_i}{\zeta}\right)^c}^{\infty} u \mathrm{e}^{-u} \mathrm{d}S_T$$
$$- \mathrm{e}^{-R_\mathrm{f}(T-t)} K_i \mathrm{e}^{-(K_i/\zeta)^c}.$$

For the put option, we have similarly:

$$\hat{P}_{it} = \mathrm{e}^{-R_\mathrm{f}(T-t)} \int_0^{K_i} (K_i - S_T) \hat{f}_t(S_T) \mathrm{d}S_T$$
$$= \mathrm{e}^{-R_\mathrm{f}(T-t)} K_i \left(1 - \mathrm{e}^{-(K_i/\zeta)^c}\right)$$
$$- \mathrm{e}^{-R_\mathrm{f}(T-t)} \zeta \int_0^{\left(\frac{K_i}{\zeta}\right)^c} u^{\frac{1}{c}} \mathrm{e}^{-u} \mathrm{d}u.$$

Note that:

$$\int_0^{u_i} u^{\frac{1}{c}} \mathrm{e}^{-u} \mathrm{d}u = \gamma\left(1 + \frac{1}{c}, u_i\right) \quad \text{and}$$
$$\int_0^{\left(\frac{K_i}{\zeta}\right)^c} u^{\frac{1}{c}} \mathrm{e}^{-u} \mathrm{d}u = \gamma\left[1 + \frac{1}{c}, \left(\frac{K_i}{\zeta}\right)^c\right]$$

and therefore

$$\hat{P}_{it} = \mathrm{e}^{-R_\mathrm{f}(T-t)} K_i \left(1 - \mathrm{e}^{-(K_i/\zeta)^c}\right)$$
$$- \mathrm{e}^{-R_\mathrm{f}(T-t)} \zeta\gamma\left[1 + \frac{1}{c}, \left(\frac{K_i}{\zeta}\right)^c\right].$$

The underlying price is then

$$\hat{S}_t = \mathrm{e}^{-R_\mathrm{f}(T-t)} E_{\mathrm{RND,W}}(S_T) \quad \text{and}$$
$$\hat{S}_t = \mathrm{e}^{-R_\mathrm{f}(T-t)} \zeta \Gamma\left(\frac{1}{c} + 1\right).$$

We can use other distributions as well. For example, several authors like to use the Burr distribution because it includes as special cases numerous and well-known distributions. In his case, we have

$$F(S_T, T) = 1 - \frac{1}{\left(1 + S_T^c\right)^q}, \; S_T \geq 0, c, q > 1;$$
$$f(S_T, T) = \frac{qcS_T^{c-1}}{\left(1 + S_T^c\right)^{q+1}}$$

or using the following notation

$$f(S_T) = \frac{\alpha\nu S_T^{\alpha-1}}{\left(S_T^\alpha + \delta\right)^{\delta+1}},$$
$$E\left(S_T^m\right) = \frac{1}{\alpha}\delta^{(m+\alpha)/\alpha} B\left(1 + \frac{m}{\alpha}, \delta - \frac{m}{\alpha}\right),$$

which can be used to fit an available data set to the distribution and optimize to obtain parameter estimates. The problem with this technique, however, is that it is mostly appropriate for an estimation of a specific risk-neutral distribution for a specific instant of time rather than to the evolution of the risk-neutral distribution over a stochastic process. Similar considerations are applied when we use the Burr distribution. In this case, for a Burr III distribution we have

$$F_{\mathrm{BR}}(S_T, T) = \left(1 - \frac{1}{1 + (S_T/\beta)^c}\right)^\alpha,$$
$$S_T \geq 0, c > 0, \alpha > 0, \beta > 0$$

while the probability distribution is

$$f(S_T, T) = \frac{c\alpha S_T^{c\alpha-1}\left(S_T^c + \beta^c\right) - c\alpha S_T^{c(\alpha-1)}}{\left(S_T^c + \beta^c\right)^{1+\alpha}}.$$

Then by simply minimizing the sum of squared differences between a model premia conditional upon the parameters of the distribution and the observed option premia, an estimate of the approximate risk-neutral distribution can be obtained (for references see [47.191, 192]).

An alternative approach consists of recovering the risk-neutral distribution associated with an asset based on the information available on its derived products. For example, under risk-neutral pricing, for a vanilla call option with exercise price K and exercise date T, we have by definition:

$$C(K,T) = \mathrm{e}^{-R_f T} \int_K (S-K) f(S)\,\mathrm{d}t\,,$$

where $f(S)$ is the underlying asset risk-neutral density function of the price at its exercise time. Note that:

$$\begin{aligned}
\frac{\partial C(K,T)}{\partial K} &= -\mathrm{e}^{-R_f T} \int_K f(S)\,\mathrm{d}t \\
&= -\mathrm{e}^{-R_f T}\,[1-F(K)]\,, \\
\frac{\partial^2 C(K,T)}{\partial K^2} &= \mathrm{e}^{-R_f T} f(S)\,, \\
f(S) &= \mathrm{e}^{R_f T} \frac{\partial^2 C(K,T)}{\partial K^2}\,,
\end{aligned}$$

where $f(S)$, the RND is explicitly stated above as the second partial derivative of the call option. Of course, although such a relationship may be of theoretical value, it has little use if there is no data that can be used to calculate the underlying risk-neutral distribution. Other approaches and techniques might therefore be needed. This is a broad and difficult area of research which we can consider here only briefly. One approach that may be used is based on the constrained maximization of entropy to determine candidate risk-neutral distributions. This is considered next.

47.6.4 The Maximum-Entropy Approach

When some characteristics, data or other information regarding the risk-neutral distribution are available, it is possible to define its underlying distribution by selecting that distribution which assumes the least, that is the distribution with the greatest variability, given the available information. One approach that allows the definition of such distributions is defined by the maximum-entropy principle.

Entropy is based essentially on a notion of randomness. Its origins are in statistical physics. Boltzmann observed that entropy relates to missing information inasmuch as it pertains to the number of alternatives which remain possible to a physical system after all the macroscopically observable information concerning it has been recorded. In this sense, information can be interpreted as that which changes a system's state of randomness (or equivalently, as that quantity which reduces entropy). For example, for a word, which has k letters assuming zeros and ones, and one two, define a sequence of k letters, $(a_0, a_1, a_2, \ldots, a_k)$,

$$a_i = \begin{cases} 1 \\ 0 \end{cases}$$

for all $i \neq j$, $a_j \neq a_i = 2$ and one j. The total number of configurations (or strings of $k+1$ letters) that can be created is N where, $N = 2^k(k+1)$. The logarithm to the base 2 of this number of configurations is the information I, or $I = \log_2 N$ and in our case, $I = k + \log_2(k+1)$. The larger this number I, the larger the number of possible configurations and therefore the larger the randomness of the word. As a further example, assume an alphabet of G symbols and consider messages consisting of N symbols. Say that the frequency of occurrence of a letter is f_i, i. e., in N symbols the letter G occurs on average $N_i = f_i N$ times. There may then be W different possible messages, where

$$W = \frac{N!}{\prod_{i=1}^{N} N_i!}\,.$$

The uncertainty of an N-symbol message is simply the ability to discern which message is about to be received. Thus,

$$\begin{aligned}
&W = \mathrm{e}^{HN} \quad \text{and} \\
&H = \lim_{N\to\infty} \frac{1}{N} \log(W) = \sum p_i \log\left(1/p_i\right),
\end{aligned}$$

which is also known as Shannon's entropy. If the number of configurations (i. e. W) is reduced, then the *information* increases. To see how the mathematical and statistical properties of entropy may be used in defining the risk-neutral distribution, we shall outline below a number of problems.

Discrimination and divergence

Consider for example two probability distributions given by $[F, G]$, one a theoretical risk-neutral distribution and another empirical expressing observed prices for example. We want to construct a measure that makes it possible to discriminate between these distributions. An attempt may be reached by using the following function we call the discrimination information [47.193]:

$$I(F,G) = \int F(x) \log \frac{F(x)}{G(x)}\,\mathrm{d}x\,.$$

In this case, $I(F, G)$ is a measure of *distance* between the distributions F and G. The larger the measure, the more we can discriminate between these distributions. For example, if $G(.)$ is a uniform distribution, then we have Shannon's measure of information. In this sense, it also provides a measure of departure from the random distribution. Selecting a distribution which has a maximum entropy (given a set of assumptions which are made explicit) is thus equivalent to the principle of insufficient reason proposed by Laplace. Thus, selecting a distribution with the largest entropy will imply a most conservative (risk wise) distribution. By the same token, we have

$$I(G, F) = \int G(x) \log \frac{G(x)}{F(x)} \mathrm{d}x$$

and the *divergence* between these two distributions is defined by

$$\begin{aligned} J(F, G) &= I(F, G) + I(G, F) \\ &= \int [F(x) - G(x)] \log \frac{F(x)}{G(x)} \mathrm{d}x \,, \end{aligned}$$

which provides a *symmetric measure* of distributions' *distance* since $J(F, G) = J(G, F)$.

For a discrete-time distribution (p, q), discrimination and divergence are given by

$$I(p, q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \,;$$

$$I(q, p) = \sum_{i=1}^{n} q_i \log \frac{q_i}{p_i}, \sum_{i=1}^{n} p_i = 1 \,,$$

$$\sum_{i=1}^{n} q_i = 1, \quad p_i \geq 0, \quad q_i \geq 0 \,,$$

$$\begin{aligned} J(p, q) &= \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} + \sum_{i=1}^{n} q_i \log \frac{q_i}{p_i} \\ &= \sum_{i=1}^{n} (p_i - q_i) \log \frac{p_i}{q_i} \,. \end{aligned}$$

For example, say that q_i, $i = 1, 2, 3, \ldots n$ is a known empirical distribution and say that p_i, $i = 1, 2, 3, \ldots n$ is a theoretical distribution given by the geometric distribution: $p_i = (n, i)^{\top} p^i (1 - p)^{n-i}$, whose parameter p we seek to estimate by minimizing the divergence, then the problem is:

$$\begin{aligned} &\min_{0 \leq p \leq 1} J(p, q) \\ &= \sum_{i=1}^{n} \left[\binom{n}{i} p^i (1 - p)^{n-i} - q_i \right] \\ &\quad \times \log \frac{\binom{n}{i} p^i (1 - p)^{n-i}}{q_i} \,, \end{aligned}$$

which can be minimized with respect to the parameter p. This approach can be generalized further to a multi-variable setting. For a bi-variate state discrete distribution, we have similarly:

$$I(p, q) = \sum_{j=1}^{m} \sum_{i=1}^{n} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) ;$$

$$J(p, q) = \sum_{j=1}^{m} \sum_{i=1}^{n} (p_{ij} - q_{ij}) \log \left(\frac{p_{ij}}{q_{ij}} \right) .$$

On other hand for, for continuous distributions, we also have

$$I(F, G) = \int \int F(x, y) \log \frac{F(x, y)}{G(x, y)} \mathrm{d}x \, \mathrm{d}y$$

as well as the divergence:

$$\begin{aligned} &J(F, G) \\ &= \int \int [F(x, y) - G(x, y)] \log \frac{F(x, y)}{G(x, y)} \mathrm{d}x \, \mathrm{d}y \,. \end{aligned}$$

This distribution may then be used to provide divergence–distance measures between empirically observed and theoretical distributions.

When the underlying process is time-varying, we have for each time period:

$$I(p, q) = \sum_{t=1}^{T} \sum_{i=1}^{n} p_{it} \log \left(\frac{p_{it}}{q_{it}} \right) \quad \text{and}$$

$$J(p, q) = \sum_{t=1}^{T} \sum_{i=1}^{n} (p_{it} - q_{it}) \log \left(\frac{p_{it}}{q_{it}} \right)$$

where, obviously,

$$\sum_{i=1}^{n} p_{it} = 1; \quad \sum_{i=1}^{n} q_{it} = 1; \quad p_{it} \geq 0, q_{it} \geq 0 \,.$$

Moments condition as well as other constraints may also be imposed, providing a *least-divergent* risk-neutral pricing approximation to the empirical (incomplete) distribution considered.

Example 47.9: Assume that a non-negative random security price $\{\theta\}$ has a known mean given by $\hat{\theta}$, the maximum-entropy distribution for a continuous state distribution is given by solution of the following optimization problem:

$$\max H = -\int_0^\infty f(\theta)\log[f(\theta)]\,\mathrm{d}\theta \quad \text{Subject to:}$$

$$\int_0^\infty f(\theta)\mathrm{d}\theta = 1,$$

$$\hat{\theta} = \int_0^\infty \theta f(\theta)\mathrm{d}\theta\,.$$

The solution of this problem, based on the calculus of variations, yields an exponential distribution. In other words,

$$f(\theta) = \frac{1}{\hat{\theta}}\mathrm{e}^{-\theta/\hat{\theta}}, \quad \theta \geq 0\,.$$

When the variance of a distribution is specified as well, it can be shown that the resulting distribution is the normal distribution with specified mean and specified variance. This approach can be applied equally when the probability distribution is discrete, bounded, and multivariate with specified marginal distributions etc. In particular, it is interesting to point out that the maximum entropy of a multivariate distribution with specified mean and known variance–covariance matrix also turns out to be multivariate normal, implying that the normal is the most random distribution that has a specified mean and a specified variance. Evidently, if we also specify leptokurtic parameters, the distribution will not be normal. Potential applications are numerous, for example, let S and V be a stock price and its volatility, each of which is assumed to have observable prices and volatility. If we apply the conditions for a risk-neutral price, we then have:

$$S(t) = \mathrm{e}^{-R_\mathrm{f}(T-t)}E_\mathrm{RN}S(T)$$
$$= \int_0^\infty\int_0 \mathrm{e}^{-R_\mathrm{f}(T-t)}S(T)f(S,V,T)\mathrm{d}V\,\mathrm{d}S\,,$$

where $f(S,V,T)$ is the probability distribution of the stock price with volatility V at time T. Adding data regarding the observed volatility at various times, prices of call and puts derived from this security, a theoretical optimization problem can be constructed that will indicate potential candidate distributions as implied risk-neutral distributions.

Example 47.10: consider the following random volatility process:

$$x_{t+1}^\alpha = x_t^\alpha + z_t; \quad z_t = \begin{cases} +3 & (0.5)(1-q) \\ +1 & (0.5)q \\ -1 & (0.5)q \\ -3 & (0.5)(1-q)\,. \end{cases}$$

A three-stage standard binomial process with probability π leads to

$$x_{t+1}^\alpha = x_t^\alpha + z_t\,;$$
$$z_t = \begin{cases} +3 & 0.5(1-q) \leftrightarrow \pi^3 \\ +1 & 0.5q \leftrightarrow 3\pi^2(1-\pi) \\ -1 & 0.5q \leftrightarrow 3\pi(1-\pi)^2 \\ -3 & 0.5(1-q) \leftrightarrow (1-\pi)^3\,. \end{cases}$$

As a result, we can calculate the probability π by minimizing the divergence J, which is given by by an appropriate choice of π

$$J = \left[\pi^3 - (0.5)(1-q)\right]\log\left(\frac{\pi^3}{0.5(1-q)}\right)$$
$$+\left[3\pi^2(1-\pi) - (0.5)q\right]\log\left(\frac{3\pi^2(1-\pi)}{0.5q}\right)$$
$$+\left[3\pi(1-\pi)^2 - (0.5)q\right]\log\left(\frac{3\pi(1-\pi)^2}{0.5q}\right)$$
$$+\left[(1-\pi)^3 - 0.5(1-q)\right]\log\left(\frac{(1-\pi)^3}{0.5(1-q)}\right)\,.$$

Example 47.11: the problem based on forward and option prices be given by:

$$\max_{f(.)}\int_0^\infty f(S)\ln\left(\frac{1}{f(S)}\right)\mathrm{d}S \quad \text{Subject to:}$$

$$\int_0^\infty f(S)\mathrm{d}S = 1; \quad F(0,T) = \int_0^\infty Sf(S)\mathrm{d}S \quad \text{and}$$

$$C_i(S,K,T) = \mathrm{e}^{-R_\mathrm{f}T}\int_0^\infty c_i(x)f(x)\mathrm{d}x,$$

$$i = 1,2,\ldots m\,,$$

where C_i is the price of the option at time t with payoff at T given by $c_i(x)$. The solution of this problem is

$$f(S) = \frac{1}{\mu} \exp\left[\lambda_0 S + \sum_{i=1}^{m} \lambda_i c_i(S)\right] \quad \text{with}$$

$$\mu = \int_0^\infty \exp\left[\lambda_0 S + \sum_{i=1}^{m} \lambda_i c_i(S)\right] \mathrm{d}S$$

which can be used as a candidate risk-neutral distribution where the parameters are to be determined based on the available data.

Example 47.12 (a maximum-entropy price process): Consider a bivariate probability distribution (or a stochastic price process) $h(x, t)$, $x \in [0, \infty)$, $t \in [a, b]$ the maximum-entropy criterion can be written as an optimization problem, maximizing the entropy as follows:

$$\max H = -\int_0^\infty \int_a^b h(x,t) \log[h(x,t)] \,\mathrm{d}t \,\mathrm{d}x \,,$$

subject to partial information regarding the distribution $h(x, t)$, $x \in [0, \infty)$, $t \in [a, b]$. Say that, at the final time b, the price of a stock is for sure X_b, while initially it is given by X_a. Further, let the average price over the relevant time interval be known and be given by $\bar{X}_{(a,b)}$, this may be translated into the following constraints:

$$h(X_a, a) = 1, \quad h(X_b, b) = 1 \quad \text{and}$$

$$\frac{1}{b-a} \int_a^b \int_0^\infty x h(x,t) \,\mathrm{d}x \,\mathrm{d}t = \bar{X}_{(a,b)}$$

which are to be accounted for in the entropy optimization problem. Of course, we can add additional constraints when more information is available. Thus, the maximum-entropy approach can be used as an *alternative rationality* for the construction of risk-neutral distributions when the burden of explicit hypotheses formulation or the justification of the model at hand is too heavy. Theoretical justifications as well as applications to finance may be found in *Avellaneda* et al. [47.194] (see also [47.195–197]). Below we consider a simple example to highlight some of the practical issues we may have to address when dealing with such problems.

Example 47.13 (engineered bond pricing): consider a Vasicek model of interest rates, fluctuating around a long-run rate α. This fluctuation is subjected to random and normal perturbations of mean zero and variance $\sigma \,\mathrm{d}t$, or.

$$\mathrm{d}r = \beta(\alpha - r)\mathrm{d}t + \sigma \,\mathrm{d}w$$

whose solution at time t when the interest rate is $r(t)$ is, as seen earlier:

$$r(u; t) = \alpha + \mathrm{e}^{-\beta(u-t)}[r(t) - \alpha] + \sigma \int_t^u \mathrm{e}^{-\beta(u-\tau)} \,\mathrm{d}w(\tau) \,.$$

In this theoretical model we might consider the parameters set $\Lambda \equiv (\alpha, \beta, \sigma)$ as determining a number of martingales (or bond prices) that obey the model above, namely bond prices at time $t = 0$ can theoretically equal the following:

$$B_{\mathrm{th}}(0, T; \alpha, \beta, \sigma) = E\left(\mathrm{e}^{\int_0^T r(u;\alpha,\beta,\sigma)\mathrm{d}u}\right) .$$

In this simple case, interest rates have a normal distribution with a mean and variance (volatility) evolution stated above and therefore $\int_0^T r(u,\alpha,\beta,\sigma)\mathrm{d}u$ also has a normal probability distribution with mean and variance give by

$$m[r(0), T] = \alpha T + \left(1 - \mathrm{e}^{-\beta T}\right) \frac{r(0) - \alpha}{\beta} \,,$$

$$v(r(0), T) = v(T) = \frac{\sigma^2}{2\beta^3}\left(4\mathrm{e}^{-\beta T} - \mathrm{e}^{-2\beta T} + 2\beta T - 3\right) .$$

Note that in these equations the variance is independent of the interest rate while the mean is a linear function of the interest, which we write as:

$$m[r(0), T] = \alpha\left(T - \frac{\left(1 - \mathrm{e}^{-\beta T}\right)}{\beta}\right) + r(0)\frac{\left(1 - \mathrm{e}^{-\beta T}\right)}{\beta} \,.$$

This property is called an affine structure as we saw earlier and is of course computationally desirable for it will allow a simpler calculation of the desired martingale. As a result, the theoretical zero-coupon bond price paying

one dollar T periods hence can be written as:

$$
\begin{aligned}
B_{\text{th}}(0, T; \alpha, \beta, \sigma) &= E\left(\mathrm{e}^{\int_0^T r(u,\alpha,\beta,\sigma)\,\mathrm{d}u}\right) \\
&= \mathrm{e}^{-m(r(0),T)+v(T)/2} \\
&= \mathrm{e}^{A(T)-r_0 D(T)}\,, \\
A(T) &= -\alpha\left(T - \frac{\left(1-\mathrm{e}^{-\beta T}\right)}{\beta}\right) \\
&\quad + \frac{\sigma^2}{4\beta^3}\left(4\mathrm{e}^{-\beta T} - \mathrm{e}^{-2\beta T}\right. \\
&\quad \left. + 2\beta T - 3\right)\,, \\
D(T) &= \frac{\left(1-\mathrm{e}^{-\beta T}\right)}{\beta}\,.
\end{aligned}
$$

Assume now that continuous series affine bond values are observed and given by $B_{\text{obs}}(0, T)$ which we write for convenience as $B_{\text{obs}}(0, T) = \mathrm{e}^{-R_T T}$. Without loss of generality we can consider the yield error term given by

$$\Delta_T = R_T - [A(T) - r_0 D(T)]$$

and thus select the parameters (i. e. the martingale) that is closest in some sense to the observed values. For example, a least-squares solution of n observed bond values yields the following optimization problem:

$$\min_{\alpha,\beta,\sigma} \sum_{i=1}^{n} (\Delta_i)^2$$

Alternatively, we can also minimize the divergence between the theoretical and the observed series. For a continuous-time function $B_{\text{obs}}(0, T) = \mathrm{e}^{-R_T T}$, we have

$$
\begin{aligned}
&\min_{\alpha,\beta,\sigma} J(\Upsilon) \\
&= \int_0^{\Upsilon} [B_{\text{th}}(0, u) - B_{\text{obs}}(0, u)] \ln\left(\frac{B_{\text{th}}(0, u)}{B_{\text{obs}}(0, u)}\right) \mathrm{d}u\,.
\end{aligned}
$$

In this case, it is easy to show that, given the continuous-time observed bond function $\left[B_{\text{obs}}(0, u), 0 \le u \le \Upsilon\right]$, the optimal parameters satisfy the following three equalities

$$
\begin{aligned}
&\int_0^{\Upsilon} \left(\frac{\partial \ln B_{\text{th}}(0, u)}{\partial\theta} [B_{\text{obs}}(0, u) - B_{\text{th}}(0, u)]\right) \\
&= \int_0^{\Upsilon} \frac{\partial B_{\text{th}}(0, u)}{\partial\theta} \ln\left(\frac{B_{\text{th}}(0, u)}{B_{\text{obs}}(0, u)}\right) \mathrm{d}u\,,
\end{aligned}
$$

where $\theta = \alpha, \beta, \sigma$. These problems can be solved numerically of course.

When the model has time-varying parameters, the problem we faced above turns out to have an infinity of unknown parameters and therefore the yield-curve estimation problem we considered above might be grossly under-specified. Explicitly, let the interest mode be defined by:

$$\mathrm{d}r(t) = \beta[\alpha(t) - r(t)]\,\mathrm{d}t + \sigma\,\mathrm{d}w$$

The theoretical bond value still has an affine structure and therefore we can write

$$
\begin{aligned}
B_{\text{th}}(t, T; \alpha(t), \beta, \sigma) &= E\left(\mathrm{e}^{\int_0^T r(u,\alpha,\beta,\sigma)\,\mathrm{d}u}\right) \\
&= \mathrm{e}^{A(t,T)-r(t)D(t,T)}\,.
\end{aligned}
$$

The integral interest rate process is still normal with mean and variance leading to

$$
\begin{aligned}
D(t, T) &= \frac{1}{\beta}\left(1 - \mathrm{e}^{-\beta(T-t)}\right)\,, \\
A(t, T) &= \int_t^T \left[\frac{1}{2}\sigma^2 D^2(s, T) - \beta\alpha(s) D(s, T)\right] \mathrm{d}s
\end{aligned}
$$

or

$$
\begin{aligned}
\frac{\mathrm{d}A(t, T)}{\mathrm{d}t} &= \alpha(t)\left(1 - \mathrm{e}^{-\beta(T-t)}\right) \\
&\quad - \frac{\sigma^2}{2\beta^2}\left(1-\mathrm{e}^{-\beta(T-t)}\right)^2\,, \quad A(T, T) = 0
\end{aligned}
$$

in which $\alpha(t)$, β, and σ are unspecified. If we equate this equation to the available bond data we will obviously have far more unknown variables than data points and therefore the yield-curve estimate will depend again on the optimization technique we use to generate the best fit functions $\alpha^*(t)$, β^*, and σ^*. Such problems can be formulated as standard problems in the calculus of variations. For example, if we consider the observed bond prices $B_{\text{obs}}(t, T)$, $t \le T < \infty$, for a specific time T and minimize the squared error, the following problem results

$$
\begin{aligned}
\min_{\alpha(u)} J(\alpha, A) = \int_0^t \Bigg[&\mathrm{e}^{A(u,T)-r(u)\left[\frac{1}{\beta}\left(1-\mathrm{e}^{-\beta(T-u)}\right)\right]} \\
&- B_{\text{obs}}(u, T)\Bigg]^2 \mathrm{d}u,
\end{aligned}
$$

$$\frac{\mathrm{d}A(u,T)}{\mathrm{d}u} = \alpha(u)\left(1-\mathrm{e}^{-\beta(T-u)}\right) - \frac{\sigma^2}{2\beta^2}\left(1-\mathrm{e}^{-\beta(T-u)}\right)^2,$$
$$A(T,T)=0$$

which can be solved by the usual techniques in optimal control. Note that this problem can be written as the linear quadratic optimization problem where we have purposely given greater weight to data observed close to time t, and given less importance to data that are farther away from the current time t, or:

$$\min_{\alpha(u)} = \int_0^t \mathrm{e}^{\upsilon u}\left[A(u,T)-c(u,T)\right]^2 \mathrm{d}u\,,$$
$$\frac{\mathrm{d}A(u,T)}{\mathrm{d}u} = \alpha(u)a(u,T)-b(u,T)\,, \quad A(T,T)=0\,,$$
$$c(u,t) = y_{\mathrm{obs}}(u,T)+r(u)\left[\frac{1}{\beta}\left(1-\mathrm{e}^{-\beta(T-u)}\right)\right],$$
$$a(u,t) = \left(1-\mathrm{e}^{-\beta(T-u)}\right);$$
$$b(u,t) = \frac{\sigma^2}{2\beta^2}\left(1-\mathrm{e}^{-\beta(T-u)}\right)^2.$$

The solution of this problem is a standard optimal control problem which may be either a boundary solution (called bang–bang, bringing the control parameter $\alpha(u)$ to an upper or lower constraint value) or that can be singular (in which case its calculation is found by tests based on a higher-order derivative). Using the deterministic dynamic programming framework, we have an optimal solution given by:

$$-\frac{\partial J}{\partial u} = \min_{\alpha(u)}\left\{\mathrm{e}^{\upsilon u}\left[A(u,T)-c(u,T)\right]^2 + \frac{\partial J}{\partial A}\left[\alpha(u)a(u,T)-b(u,T)\right]\right\}$$

On a singular strip, $\frac{\partial J}{\partial A}=0$ where $a(u,t)\neq 0$ and thus, in order to calculate $\alpha(u)$, we can proceed by a change of variables and transforming the original control problem into a linear quadratic control problem that can be solved by the standard optimal control methods. Explicitly, set:

$$y(u) = \mathrm{e}^{\nu u/2}\left[A(u,T)-c(u,T)\right],$$
$$\frac{\mathrm{d}w(u)}{\mathrm{d}u} = \alpha(u) \quad \text{and}$$
$$z(u) = y(u)-\mathrm{e}^{\nu u/2}a(u,T)w(u).$$

Thus, the problem objective is reduced to

$$\min_{\alpha(u)} = \int_0^t \left[z(u)+e^{\nu u/2}a(u,T)w(u)\right]^2 \mathrm{d}u$$

while the constraint is

$$\frac{\mathrm{d}A(u,T)}{\mathrm{d}u} = \alpha(u)a(u,T)-b(u,T)\,,$$
$$z(u)+\mathrm{e}^{\nu u/2}a(u,T)w(u) = \mathrm{e}^{\nu u/2}\left[A(u,T)-c(u,T)\right];$$
$$\frac{\mathrm{d}w(u)}{\mathrm{d}u} = \alpha(u)\,.$$

After some elementary manipulations, we have

$$\dot{z}(u) = \frac{\upsilon}{2}z(u)-\mathrm{e}^{\nu u/2}\dot{a}(u,T)w(u) - \mathrm{e}^{\nu u/2}\left[b(u,T)-\mathrm{e}^{\nu u/2}\dot{c}(u,T)\right].$$

This defines a linear quadratic cost-control problem

$$\min_{w(u)} = \int_0^t \left[z(u)+\mathrm{e}^{\nu u/2}a(u,T)w(u)\right]^2 \mathrm{d}u\,.$$

Subject to:

$$\frac{\mathrm{d}z(u)}{\mathrm{d}u} = \frac{\upsilon}{2}z(u)-\mathrm{e}^{\nu u/2}\dot{a}(u,T)w(u) - \mathrm{e}^{\nu u/2}\left[b(u,T)-\mathrm{e}^{\nu u/2}\dot{c}(u,T)\right].$$

Inserting the original problem parameters we have:

$$\min_{w(u)} \int_0^t \left[z(u)+\mathrm{e}^{\nu u/2}\left\{1-\mathrm{e}^{-\beta(T-u)}\right\} \times w(u)\right]^2 \mathrm{d}u. \quad \text{Subject to:}$$
$$\frac{\mathrm{d}z(u)}{\mathrm{d}u} = \dot{z}(u) = \frac{\upsilon}{2}z(u)+\beta\,\mathrm{e}^{-\beta(T-u-\nu u/2\beta)}w(u) - \mathrm{e}^{\nu u/2}\left[\frac{\sigma^2}{2\beta^2}\left(1-\mathrm{e}^{-\beta(T-u)}\right)^2 - \mathrm{e}^{\nu u/2}\dot{c}(u,T)\right],$$
$$\frac{\mathrm{d}c(u)}{\mathrm{d}u} = \dot{c}(u,t) = \dot{y}_{\mathrm{obs}}(u,T) + \dot{r}(u)\left[\frac{1}{\beta}\left(1-\mathrm{e}^{-\beta(T-u)}\right)\right] - r(u)\,\mathrm{e}^{-\beta(T-u)},$$

which is a control problem linear in the state and in the control with a quadratic objective. As a result, the solution for the control $w(u)$ is of the linear feedback form

$$w(u) = Q(u) + S(u)z(u) \quad \text{or}$$
$$\alpha(u) = \dot{Q}(u) + S(u)\dot{z}(u) + \dot{S}(u)z(u)\,.$$

Finally, when bond data are available over multiple periods dates T, the optimal control problem we have considered above can be extended by solving

$$\min_{\alpha(u)} \sum_{j=1}^{N_T} \int_0^t e^{\nu u} \left[A(u, T_j) - c(u, T_j)\right]^2 \mathrm{d}u$$

$$\frac{\mathrm{d}A(u, T_j)}{\mathrm{d}u} = \alpha(u)a(u, T_j) - b(u, T_j)\,,$$
$$A(T_j, T_j) = 0$$

and in continuous time:

$$\min_{\alpha(u)} \int_0^\infty \int_0^t \mathrm{e}^{\nu u} \left[A(u, T) - c(u, T)\right]^2 \mathrm{d}T\,\mathrm{d}u\,,$$
$$\frac{\partial A(u, T)}{\partial u} = \alpha(u)a(u, T) - b(u, T)\,,$$
$$A(T, T) = 0\,.$$

The solution of these problems are then essentially numerical problems, however.

References

47.1 C. S. Tapiero: *Risk and Financial Management: Mathematical and Computationl Concepts* (Wiley, London, March 2005)

47.2 C. S. Tapiero: Risk management. In: *Encyclopedia on Actuarial and Risk Management*, ed. by E. J. Teugels, B. Sundt (Wiley, New York, London 2004)

47.3 P. Artzner, F. Delbaen, J. M. Eberand, D. Heath: Thinking coherently, RISK **10**, 68–71 (1997)

47.4 P. Artzner: Application of coherent measures to capital requirements in insurance, North Am. Actuar. J. **3**(2), 11–25 (1999)

47.5 P. Artzner, F. Delbaen, J. M. Eber, D. Heath: Coherent risk measure, Math. Finance **9**, 203–228 (1999)

47.6 H. Raiffa, R. Schlaiffer: *Applied Statistical Decision Theory* (Division of Research, Graduate School of Business, Harvard University, Boston 1961)

47.7 C. Alexander: *Risk Management and Analysis*, Vol. 1, 2 (Wiley, New York 1998)

47.8 F. Basi, P. Embrechts, M. Kafetzaki: Risk management and quantile estimation. In: *Practical Guide to Heavy Tails*, ed. by R. Adler, R. Feldman, M. Taqqu (Birkhauser, Boston 1998) pp. 111–130

47.9 S. Beckers: A survey of risk measurement theory and practice. In: *Handbook of Risk Management and Analysis*, ed. by C. Alexander (Wiley, New York 1996)

47.10 P. P. Boyle: *Options and the Management of Financial Risk* (Society of Actuaries, New York 1992)

47.11 Neil A. Doherty: *Integrated Risk Management: Techniques and Strategies for Managing Corporate Risk* (McGraw–Hill, New York 2000)

47.12 J. E. Ingersoll, Jr.: *Theory of Financial Decision Making* (Rowman and Littlefield, New Jersey 1987)

47.13 P. Jorion: *Value at Risk: The New Benchmark for Controlling Market Risk* (McGraw–Hill, Chicago 1997)

47.14 P. Embrechts, C. Klupperberg, T. Mikosch: *Modelling Extremal Events in Insurance and Finance* (Springer, Berlin Heidelberg New York 1997)

47.15 C. Gourieroux, J. P. Laurent, O. Scaillet: Sensitivity analysis of values at risk, J. Empirical Finance **7**, 225–245 (2000)

47.16 S. Basak, A. Shapiro: Value-at-risk-based risk management: Optimal policies and asset prices, Rev. Financial Stud. **14**, 371–405 (2001)

47.17 D. E. Bell: Risk, return and utility, Manage. Sci. **41**, 23–30 (1995)

47.18 Eugene F. Fama: The cross-section of expected stock returns, J. Finance **47**, 427–465 (1992)

47.19 K. J. Arrow: Aspects of the theory of risk bearing, YRJO Jahnsson Lectures (1963), also in 1971 *Essays in the Theory of Risk Bearing*, Markham Publ. Co., Chicago, Ill

47.20 J. Y. Campbell: Asset pricing at the Millennium, J. Finance **LV, 4**, 1515–1567 (2000)

47.21 J. C. Cox, S. A. Ross: A survey of some new results in financial option pricing theory, J. Finance **31**, 383–402 (1978)

47.22 D. Duffie: *Security Markets: Stochastic Models* (Academic, New York 1988)

47.23 D. Duffie: *Dynamic Asset Pricing Survey* (Working Paper, Stanford University 2002)

47.24 R. C. Merton: *Continuous Time Finance* (M. A. Blackwell, Cambridge 1990)

47.25 R. A. Jarrow: *Finance Theory* (Prentice Hall, Englewood Cliffs, N.J. 1988)

47.26 J. M. Bismut: Growth and intertemporal allocation of risks, J. Econ. Theory **10**, 239–257 (1975)

47.27 J. M. Bismut: An introductory approach to duality in optimal stochastic control, SIAM Rev. **20**, 62–78 (1978)
47.28 W. A. Brock, M. J. P. Magill: Dynamics under uncertainty, Econometrica **47**, 843–868 (1979)
47.29 J. B. Caouette, E. I. Altman, P. Narayanan: *Managing Credit Risk: The Next Great Financial Challenge* (Wiley, New York 1998)
47.30 D. Cossin, H. Pirotte: *Advanced Credit Risk Analysis: Financial Approaches and Mathematical Models to Assess, Price and Manage Credit Risk* (Wiley, New York 2001)
47.31 J. Cox, M. Rubinstein: *Options Markets* (Prentice Hall, Englewood Cliffs, N. J. 1985)
47.32 J. C. Hull: *Options, Futures and Other Derivatives*, 4th edn. (Prentice Hall, Englewood Cliffs, N. J. 2000)
47.33 A. G. Malliaris, W. A. Brock: *Stochastic methods in Economics and Finance* (North Holland, Amsterdam 1982)
47.34 Harry M. Markowitz: *Portfolio Selection; Efficient Diversification of Investments* (Wiley, New York 1959)
47.35 Y. A. Bergman: Time preference and capital asset pricing models, J. Financial Econ. **14**, 145–159 (1985)
47.36 R. S. Dembo: Scenario optimization, Algorithmics Inc. Research Paper **89**(01) (1989)
47.37 R. S. Dembo: Scenario immunization. In: *Financial Optimization*, ed. by S. A. Zenios (Cambridge Univ. Press, London 1993)
47.38 D. Kreps: A representation theorem for preference for flexibility, Econometrica **47**, 565–577 (1979)
47.39 J. M. Harrison, S. R. Pliska: Martingales and stochastic integrals with theory of continuous trading, Stoch. Proc. Appl. **11**, 261–271 (1981)
47.40 I. Karatzas, S. E. Shreve: *Methods of mathematical finance* (Springer, New York 1998)
47.41 I. Karatzas, S. Shreve: Methods of mathematical finance, Stochastic Modelling and Applied Probability, 159–196 (1999)
47.42 D. R. Cox, H. D. Miller: *The Theory of Stochastic Processes* (Wiley, New York 1965)
47.43 H. U. Gerber: *An Introduction to Mathematical Risk Theory* (University of Penn., Philadelphia 1979) Monograph No. 8, Huebner Foundation
47.44 S. M. Ross: *Applied Probability Models with Optimization Applications* (Holden-Day, San Fransisco 1970)
47.45 S. M. Ross: *Stochastic Processes* (Wiley, New York 1982)
47.46 S. M. Ross: *Introduction to Stochastic Dynamic Programming* (Academic, New York 1983)
47.47 C. S. Tapiero: *Applied Stochastic Models and Control in Management* (North Holland, New York 1988)
47.48 C. S. Tapiero: *Applied Stochastic Control for Finance and Insurance* (Kluwer, Dordrecht 1998)
47.49 K. J. Arrow: Le role des valeurs boursieres pour la repartition la meilleur des risques, Econometric, Colloquia International due CNRS **40**, 41–47 (1953), in English in The role of securities in the optimal allocation of risk bearing, *Review of Economic Studies*, **31**, 91–96, 1963
47.50 D. Duffie: *Dynamic Asset Pricing Theory* (Princeton University Press, Princeton, New Jersey 1992)
47.51 J. Muth: Rational expectations and the theory of price movements, Econometrica **29**, 315–335 (1961)
47.52 M. Magill, M. Quinzii: *Theory of Incomplete Markets*, Vol. 1 (MIT Press, Boston 1996)
47.53 E. F. Fama: Efficient capital markets: A review of theory and empirical work, J. Finance **25**, 383–417 (1970)
47.54 R. E. Lucas: Asset prices in an exchange economy, Econometrica **46**, 1429–1445 (1978)
47.55 J. M. Harrison, D. M. Kreps: Martingales and arbitrage in multiperiod security markets, J. Econ. Theory (1979)
47.56 R. M. Capocelli, L. M. Ricciardi: On the inverse of the first passage time probability problem, J. Appl. Probab. **9**, 270–287 (1972)
47.57 T. J. Sargent: *Macroeconomic Theory* (Academic, New York 1979)
47.58 A. Bensoussan, J. L. Lions: *Controle Impulsionnel et Inequations Quasi-Variationnelles* (Dunod, Paris 1979)
47.59 A. Bensoussan, C. S. Tapiero: Impulsive control in management: Prospects and applications, J. Optim. Theory Appl. **37**, 419–442 (1982)
47.60 D. A. Darling, A. J. F. Siegert: The first passage time for a continuous Markov process, Ann. Math. Stat. **24**, 624–639 (1953)
47.61 W. F. Sharpe: Capital asset prices: A theory of market equilibrium under risk, J. Finance **19**, 425–442 (1964)
47.62 E. F. Fama, M. H. Miller: *The Theory of Finance* (Holt Rinehart and Winston, New York 1972)
47.63 E. F. Fama: The CAPM is wanted, dead or alive, J. Finance **51**, 1947 (Dec 1996)
47.64 C. Stein: Estimation of the mean of a multivariate normal distributions, Proc. Prague Symposium, Asymptotic Statistics (September 1973)
47.65 R. Roll: Asset, money and commodity price inflation under uncertainty, J. Money Credit Banking **5**, 903–923 (1973)
47.66 R. J. B. Wets, S. Bianchi, L. Yang: Serious Zero-Curve, (2002) www.episolutions.com
47.67 K. C. Chan, G. A. Karolyi, F. A. Longstaff, A. B. Sanders: An empirical comparison of alternative models of the short term interest rate, J. Finance **47**, 1209–1227 (1992)
47.68 J. D. Duffie, R. Kan: A yield-factor model of interest rates, Math. Finance **6**, 379–406 (1996)
47.69 D. C. Heath, R. A. Jarrow, A. Morton: Bond pricing and the term structure of interest rates: A new methodology for contingent claim valuation, Econometrica **60**, 77–105 (1992)
47.70 C. R. Nelson, A. F. Siegel: Parsimonious modeling of the yield curve, J. Bus. **60**, 473–489 (1987)

47.71 D. Filipovic: A note on the Nelson–Siegel family, Math. Finance **9**, 349–359 (1999)
47.72 D. Filipovic: Exponential-ploynomial families and the term structure of interest rates, Bernoulli **6**, 1–27 (2000)
47.73 D. Filipovic: Consistency problems for Heath–Jarrow–Morton interest rate models. In: *Lecture Notes in Mathematics*, Vol. 1760, ed. by J.-M. Morel, F. Takens, B. Teissier (Springer, Berlin Heidelberg New York 2001)
47.74 R. C. Merton: On the pricing of corporate debt: The risk structure of interest rates, J. Finance **29**, 449–470 (1974)
47.75 G. R. Duffee: The relation between treasury yields and corporate bond yield spreads, J. Finance **53**, 2225–2241 (1998)
47.76 G. R. Duffee: Estimating the price of default risk, Rev. Financial Stud. **12**, 197–226 (1999)
47.77 D. Duffie, K. Singleton: An econometric model of the term structure of interest rate wap yield, J. Finance **52**, 1287–1321 (1997)
47.78 D. Duffie, K. Singleton: Modeling term structures of defaultable bonds, Review Financial Stud. **12**, 687–720 (1999)
47.79 E. Elton, M. J. Gruber, D. Agrawal, C. Mann: Explaining the rate spread on corporate bonds, J. Finance **56**, 247–278 (2001)
47.80 K. O. Kortanek, V. G. Medvedev: *Building and Using Dynamic Interest Rate Models* (Wiley Finance. John Wiley & Sons Ltd., London 2001)
47.81 K. O. Kortanek: Comparing the Kortanek & Medvedev GP approach with the recent wets approach for extracting the zeros (April 26, 2003)
47.82 F. Delbaen, S. Lorimier: Estimation of the yield curve and forward rate curve starting from a finite number of observations, Insurance: Math. Econ. **11**, 249–258 (1992)
47.83 K. J. Adams, D. R. Van Deventer: Fitting yield curves and forward rate curves with maximum smoothness, J. Fixed Income, 52–62 (1994)
47.84 M. Buono, R. B. Gregoru-Allen, U. Yaari: The efficacy of term structure estimation techniques: A Monte Carlo study, J. Fixed Income **1**, 52–59 (1992)
47.85 O. A. Vasicek, H. G. Fong: Term structure modeling using exponential splines, J. Finance **37**, 339–356 (1977)
47.86 G. S. Shea: Term structure estimation with exponential splines, J. Finance **40**, 339–356 (1988)
47.87 M. Friedman, L. J. Savage: The utility analysis of choices involving risk, J. Polit. Econ. **56** (August 1948)
47.88 M. J. Brennan, E. S. Schwartz: A continuous time approach to the pricing of corporate bonds, J. Banking Finance **3**, 133–155 (1979)
47.89 J. D. Duffie, D. Fillipovic, W. Schachermayer: Affine processes and applications in finance, Ann. Appl. Probab. **13**, 19–49 (2003)
47.90 J. Hull, A. White: The pricing of options on assets with stochastic volatilitie, J. Finance **42**, 281–300 (1987)
47.91 J. C. Cox, J. E. Ingersoll, S. A. Ross: A theory of the term structure of interest rates, Econometrica **53**, 385–407 (1985)
47.92 R. Jarrow, S. Turnbull: Pricing derivatives on financial securities subject to credit risk, J. Finance **50**, 53–86 (1995)
47.93 R. A. Jarrow, D. Lando, S. Turnbull: A Markov model for the term structure of credit spreads, Rev. Financial Stud. **10**, 481–523 (1997)
47.94 D. Lando: Some elements of rating-based credit risk modeling. In: *Advanced Fixed-Income Valuation Tools*, ed. by N. Jegadeesh, B. Tuckman (Wiley, New York 2000)
47.95 F. Longstaff, E. Schwartz: A simple approach to valuing risky fixed and floating rate debt, J. Finance **50**, 789–819 (1995)
47.96 O. A. Vasicek: An equilibrium characterization of the term structure, J. Financial Econ. **5**, 177–188 (1977)
47.97 F. Black, M. Scholes: The pricing of options and corporate liabilities, J. Polit. Econ. **81**, 637–659 (1973)
47.98 M. J. Brennan: The pricing of contingent claims in discrete time models, The J. Finance **1**, 53–63 (1979)
47.99 J. C. Cox, M. Rubenstein: *Options Markets* (Prentice Hall, Englewood Cliffs, N. J. 1985)
47.100 J. C. Cox, J.E. Ingersoll jr., S. A. Ross: The relation between forward prices and futures prices, J. Financial Econ. **9**, 321–346 (1981)
47.101 J. C. Cox, S. A. Ross, M. Rubenstein: Option pricing approach, J. Financial Econ. **7**, 229–263 (1979)
47.102 R. C. Merton: Theory of rational option pricing, Bell J. Econ. Manage. Sci. **4**, 141–183 (1973)
47.103 S. Pliska: A stochastic calculus model of continuous trading: Optimal portfolios, Math. Oper. Res. **11**, 371–382 (1986)
47.104 S. A. Ross: Options and efficiency, Quarterly J. Econ. **90** (1976)
47.105 A. Ross: The arbitrage theory of capital asset pricing, J. Econ. Theory **13**, 341–360 (1976)
47.106 C. W. Smith: Option pricing: A review, J. Financial Econ. **3**, 3–51 (1976)
47.107 M. Avellenada: *Course Notes* (Courant Institue of Mathematics, New York University, New York 2001)
47.108 J. C. Cox, S. A. Ross: The valuation of options for alternative stochastic processes, J. Financial Econ., 145–166 (1976)
47.109 A. Bensoussan: *Stochastic Control by Functional Analysis Method* (North Holland, Amsterdam 1982)
47.110 A. Bensoussan, M. Hazewinkel (Ed.): On the theory of option pricing, ACTA Applicandae Mathematica **2**(2), 139–158 (1984)
47.111 P. Carr, R. Jarrow, R. Myneni: Alternative characterizations of American Put options, Math. Finance **2**, 87–106 (1992)

47.112 R. Jarrow, A. Rudd: Approximate option valuation for arbitrary stochastic processes, J. Financial Econ. **10**, 347–369 (1982)
47.113 A. Bensoussan, H. Julien: Option pricing, in a market with friction. In: *Stochastic Analysis and Applications* (1998)
47.114 A. Bensoussan, H. Julien: On the pricing of contingent claims with friction, Math. Finance **10**, 89–108 (2000)
47.115 S. D. Jacka: Optimal stopping and the American Put, J. Math. Finance **1**, 1–14 (1991)
47.116 J. Wiggins: Option values under stochastic volatility: Theory and empirical estimates, J. Financial Econ. **5**, 351–372 (1987)
47.117 J. P. Fouque, G. Papanicolaou, K. R. Sircar: *Stochastic Volatility* (Cambridge Univ. Press, Cambridge 2000)
47.118 K. Ramaswamy, D. Nelson: Simple binomial processes as diffusion approximations in financial models, Rev. Financial Stud. **3**(3), 393–430 (1990)
47.119 J. P. Bouchaud, M. Potters: *Théorie des Risques Financiers* (Aléa-Saclay/Eyrolles, Paris 1997)
47.120 B. Dupire: Pricing with a smile, RISK (January 1994)
47.121 R. Merton: Option pricing when underlying stock returns are discontinuous, J. Financial Econ. **3**, 125–144 (1976)
47.122 C. Ball, W. Torous: On jumps in common stock prices and their impact on call option price, J. Finance **40**, 155–173 (1985)
47.123 H. Cho, K. Lee: An extension of the three jump process models for contingent claim valuation, J. Derivatives **3**, 102–108 (1995)
47.124 V. Naik, M. Lee: General equilibrium pricing of options on the market portfolio with discontinuous returns, Rev. Financial Stud. **3**, 493–521 (1990)
47.125 K. Amin: Jump diffusion option valuation in discrete time, J. Finance **48**, 1833–1863 (1993)
47.126 K. I. Amin, V. K. Ng: Option valuation with systematic stochastic volatility, J. Finance **48**, 881–909 (1993)
47.127 D. E. Bell: Regret in decision making under uncertainty, Oper. Res. **30**, 961–981 (1982)
47.128 D. E. Bell: Disappointment in decision making under uncertainty, Oper. Res. **33**, 1–27 (1985)
47.129 P. C. Fishburn: *Nonlinear Preference and Utility Theory* (Johns Hopkins, Baltimore 1988)
47.130 F. Gul: A theory of disappointment aversion, Econometrica **59**, 667–686 (1991)
47.131 G. Loomes, R. Sugden: Regret theory: An alternative to rational choice under uncertainty, Econ. J. **92**, 805–824 (1982)
47.132 G. Loomes, R. Sugden: Some implications of a more general form of regret theory, J. Econ. Theory **41**, 270–287 (1987)
47.133 M. J. Machina: Choice under uncertainty: Problems solved and unsolved, J. Econ. Perspect. **1**, 121–154 (1987)
47.134 R. Sugden: An axiomatic foundation of regret theory, J. Econ. Theory **60**, 150–180 (1993)
47.135 K. J. Arrow: Risk perception in psychology and in economics, Econ. Inquiry **20**(1), 1–9 (January 1982)
47.136 M. Allais: Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'ecole americaine, Econometrica **21**, 503–546 (1953)
47.137 M. Allais: The foundations of a positive theory of choice involving risk and a criticism of the postulates and axioms of the American school. In: *Expected Utility Hypothesis and the Allais Paradox*, ed. by M. Allais, O. Hagen (Reidel, Dordrecht 1979)
47.138 D. Ellsberg: Risk, ambuguity and the Savage axioms, Q. J. Econ. **75**(4), 643–669 (November 1961)
47.139 M. Friedman, L. J. Savage: The expected utility hypothesis and the measurability of utility, J. Polit. Econ. **60**(6), 463–486 (December 1952)
47.140 M. Rabin: Psychology and economics, J. Econ. Lit. **36**, 11–46 (1998)
47.141 M. J. Machina: Expected utility analysis without the independence axiom, Econometrica **50**(2), 277–323 (March 1982)
47.142 D. Kahnemann, A. Tversky: Prospect theory: An analysis of decision under risk, Econometrica **47**(2), 263–292 (March 1979)
47.143 J. Hirschleifer: Where are we in the theory of information, Am. Econ. Rev. **63**, 31–39 (1970)
47.144 J. Hirschleifer, J. G. Riley: The analysis of uncertainty and information: An expository survey, J. Econ. Lit. **17**, 1375–1421 (1979)
47.145 G. Akerlof: The market for lemons: Quality uncertainty and the market mechanism, Quarter. J. Econ. **84**, 488–500 (1970)
47.146 B. Holmstrom: Moral hazard and observability, Bell J. Econ. **10**(1), 74–91 (1979)
47.147 E. E. Peter: *Chaos and Order in Capital Markets* (Wiley, New York 1995)
47.148 R. E. Kalman: Randomness reexamined, Modeling Identif. Control **15**(3), 141–151 (1994)
47.149 M. Born: Nobel lecture. In: *Les Prix Nobel* (Nobel Foundation, Stockholm 1954)
47.150 J. Beran: *Statistics for Long-Memory Processes* (Chapman Hall, London 1994)
47.151 S. C. Blank: 'Chaos' in futures markets? A nonlinear dynamical analysis, J. Futures Markets **11**, 711–728 (1991)
47.152 W. A. Brock, D. A. Hsieh, D. LeBaron: *Nonlinear Dynamics, Chaos and Instability: Statistical Theory and Economic Evidence* (MIT Press, Cambridge, Mass 1991)
47.153 W. A. Brock, P. J. de Lima: Nonlinear time series, complexity theory and finance. In: *Statistical Methods in Finance*, Handbook of Statistics, Vol. 14, ed. by G. Maddala, C. Rao (North Holland, Amsterdam 1996)

47.154 D. A. Hsieh: Chaos and nonlinear dynamics application to financial markets, J. Finance **46**, 1839–77 (1991)
47.155 B. LeBaron: Chaos and nonlinear forecastability in economics and finance, Phil. Trans. R. Soc. London **A 348**, 397–404 (1994)
47.156 J. A. Scheinkman, B. LeBaron: Nonlinear dynamics and stock returns, J. Bus. **62**, 311–337 (1989)
47.157 J. A. Scheinkman: Nonlinear dynamics in economics and finance, Phil. Trans. R. Soc. London **346**, 235–250 (1994)
47.158 J. P. Imhoff: On the range of Brownian motion and its inverse process, Ann. Prob. **13**(3), 1011–1017 (1985)
47.159 B. Mandelbrot: Statistical methodology for nonperiodic cycles: From the covariance to R/S analysis, Ann. Econ. Social Measure **1**, 259–290 (1972)
47.160 B. Mandelbrot, J. Van Ness: Fractional Brownian motion, fractional noises and applications, SIAM Rev. **10**, 422–437 (1968)
47.161 B. Mandelbrot, M. Taqqu: Robust R/S analysis of long run serial correlation, Bull. Int. Stat. Inst. **48**(Book 2), 59–104 (1979)
47.162 M. T. Green, B. Fielitz: Long term dependence in common stock returns, J. Financial Econ. **4**, 339–349 (1977)
47.163 D. R. Cox: Long range dependence, nonlinearity and time irreversibilit, J. Time Series Anal. **12**(4), 329–335 (1991)
47.164 M. Frank, T. Stengos: Chaotic dynamics in economic time serie, J. Econ. Surveys **2**, 103–133 (1988)
47.165 M. T. Green, B. Fielitz: Long term dependence and least squares regression in investment analysis, Manage. Sci. **26**(10), 1031–1038 (October 1980)
47.166 H. E. Hurst: Long-term storage capacity of reservoirs, Trans. Am. Soc. Civil Eng., 770–808 (1951)
47.167 J. P. Imhoff: A construction of the Brownian motion path from BES (3) pieces, Stoch. Processes Appl. **43**, 345–353 (1992)
47.168 M. S. Taqqu: A bibliographical guide to self similar processes and long range dependence. In: *Dependence in Probability and Statistics*, ed. by E. Eberlein, M. S. Taqqu (Birkhuser, Boston 1986) pp. 137–165
47.169 B. Mandelbrot: When can price be arbitraged efficiently? A limit to the the validity of the random walk and Martingale models, Rev. Econ. Stat. **53**, 225–236 (1971)
47.170 A. W. Lo: Long term memory in stock market prices, Econometrica **59**, 1279–1313 (5, September 1992)
47.171 Andrew W. Lo: Fat tails, long memory and the stock market since 1960's, Econ. Notes **26**, 213–245 (1997)
47.172 T. H. Otway: Records of the Florentine proveditori degli cambiatori: An example of an antipersistent time series in economics, Chaos Solitons Fractals **5**, 103–107 (1995)
47.173 G. Booth, F. Kaen, P. Koveos: R/S analysis of foreign exchange rates under two international monetary regimes, J. Monetary Econ **10**, 4076415 (1982)
47.174 F. Diebold, G. Rudebusch: Long memory and persistence in aggregate output, J. Monetary Econ. **24**, 189–209 (1989)
47.175 F. Diebold, G. Rudebusch: On the power of the Dickey-Fuller test against fractional alternative, Econ. Lett. **35**, 155–160 (1991)
47.176 H. G. Fung, W. C. Lo: Memory in interest rate futures, J. Futures Markets **13**, 865–873 (1993)
47.177 Y. W. Cheung: Long memory in foreign exchange rate, J. Bus. Econ. Stat. **11**, 93–101 (1993)
47.178 H. G. Fung, W. C. Lo, John E. Peterson: Examining the dependency in intra-day stock index futures, J. Futures Markets **14**, 405–419 (1994)
47.179 P. Vallois: On the range process of a Bernoulli random walk. In: *Proceedings of the Sixth International Symposium on Applied Stochastic Models and Data Analysis*, Vol. 2, ed. by J. Janssen, C. H. Skiadas (World Scientific, Singapore 1995) pp. 1020–1031
47.180 P. Vallois: The range of a simple random walk on Z, Adv. Appl. Prob. **28**, 1014–1033 (1996)
47.181 P. Vallois, C. S. Tapiero: The range process in random walks: Theoretical results and applications. In: *Adv. Comput. Econ.*, ed. by H. Ammans, B. Rustem, A. Whinston (Kluwer Publications, Dordrecht 1996)
47.182 P. Vallois, C. S. Tapiero: Run length statistics and the Hurst exponent in random and birth-death random walk, Chaos Solutions Fractals **7**(9), 1333–1341 (September 1996)
47.183 P. Vallois, C. S. Tapiero: The inter-event range process in birth death random walks, Appl. Stoch. Models Bus. Ind. **17**(3), 231–306 (2001)
47.184 C. S. Tapiero, P. Vallois: Range reliability in random walks, Math. Methods Oper. Res. **45**, 325–345 (1997)
47.185 Y. Ait-Sahalia, A. Lo: Nonparametric estimation of state price densities implicit in finncial asset prices, NBER, Working Paper No. 5351 (1995)
47.186 B. Bahra: Implied risk neutral probability density functions from prices, Bank of England, Working Paper No. 66 (1997)
47.187 A. M. Malz: Estimating the probability distribution of the future exchange rate from option prices, J. Derivatives **5**, 18–36 (1997)
47.188 R. R. Bliss, N. Panigirtzoglou: Option implied risk aversion estimates, Federal Reserve Bank of Chicago, Working Paper No. 15 (2001)
47.189 M. Rubinstein: Implied binomial trees, J. Finance **69**, 771–818 (July 1994)
47.190 J. C. Jackwerth, M. Rubinstein: Recovering probability distributions from contemporaneous security prices, J. Finance **51**, 1611–1631 (1996)
47.191 R. N. Rodriguez: A guide to the Burr type XII distributions, Biometrika **64**, 129–34 (1977)
47.192 P. R. Tadikamalla: A look at the Burr and related distributions, Int. Stat. Rev. **48**, 337–44 (1980)
47.193 S. Kullback: *Information Theory* (Dover, New York 1959)

47.194 M. Avellenada, C. Friedan, R. Holmes, D. Samperi: *Calibraing Volatility Surfaces via Relative Entropy Minimization* (Courant Institute of Mathematics, New York 2002)
47.195 P.W. Buchen, M. Kelly: The maximum entropy distribution of an asset inferred from option prices, J. Financial Quant. Anal. **31**, 143–159 (1996)
47.196 L. Gulko: The entropy theory of bond option pricing, Yale University Working paper (1995)
47.197 L. Gulko: The entropy theory of stock option pricing, Yale University Working paper (1996)

48. Statistical Management and Modeling for Demand of Spare Parts

In recent years increased emphasis has been placed on improving decision making in business and government. A key aspect of decision making is being able to predict the circumstances surrounding individual decision situations. Examining the diversity of requirements in planning and decision-making situations, it is clearly stated that no single forecasting methods or narrow set of methods can meet the needs of all decision-making situations. Moreover, these methods are strongly dependent on factors such as data quantity, pattern and accuracy, that reflect their inherent capabilities and adaptability, such as intuitive appeal, simplicity, ease of application and, not least, cost.

Section 48.1 deals with the placement of the demand-forecasting problem as one of biggest challenge in the repair and overhaul industry; after this brief introduction Sect. 48.2 summarizes the most important categories of forecasting methods; Sects. 48.3–48.4 approach the forecast of spare parts firstly as a theoretical construct, although some industrial applications and results are added from field training, as in many other parts of this chapter.

Section 48.5 undertakes the question of optimal stock level for spare parts, with particular regard to low-turnaround-index (LTI) parts conceived and designed for the satisfaction of a specific customer request, by the application of classical Poisson methods of minimal availability and minimum cost; similar considerations are drawn and compared in Sect. 48.6, which deals with models based on the binomial distribution. An innovative extension of binomial models based on the total cost function is discussed in Sect. 48.7. Finally Sect. 48.8 adds the Weibull failure-rate function to the analysis of the LTI spare-parts stock level in a maintenance system with declared wear conditions.

48.1 The Forecast Problem for Spare Parts

Demand forecasting is one of the most crucial issues for inventory management. Forecasts, which form the basis for the planning of inventory levels, are probably the biggest challenge in the repair and overhaul industry.

An example can be seen in the airline industry, where a common problem is the need to forecast short-term demand with the highest possible degree of accuracy. The high cost of modern aircraft and the expense of re-

pairable spares, such as aircraft engines and avionics, contribute significantly to the considerable total investment of many airline operators. These parts, although required with low demand, are critical to operation and their unavailability can lead to excessive downtime costs.

This problem is absolutely relevant in case of intermittent demand. Demand for an item is classified as *intermittent* when it is irregular and sporadic. This type of demand, typical for a large number of spare parts, is very difficult to predict. This complicates efficient management and control of the inventory system, which requires an acceptable balance between inventory costs on one hand and stock-outs on the other. Inventory management models require accurate forecasts in order to achieve this balance.

We can explicitly consider both the pattern and size of demand as it occurs in order to classify demand patterns into four categories [48.1], as follows:

- *intermittent demand*, which appears to be random, with many time periods having no demand,
- *erratic demand*, which is (highly) variable, the erratic nature relating to the size of demand rather than the demand per unit time period,
- *slow moving* (*smooth*) demand, which also occurs at random, with many time periods having no demand. Demand, when it occurs, is for single or very few units,
- *lumpy demand*, which likewise seems random, with many time periods having no demand. Moreover demand, when it occurs, is (highly) variable. The lumpy concept corresponds to an extremely irregular demand, with great differences between each period's requirements and with a large number of periods with zero requirements.

Traditionally the characteristics of intermittent demand are derived from two parameters: the average interdemand interval (ADI) and the coefficient of variation (CV). ADI measures the average number of time periods between two successive demands and CV represents the standard deviation of requirements divided by the average requirement over a number of time periods:

$$\mathrm{CV} = \frac{\sqrt{\frac{\sum_{i=1}^{n}(\varepsilon_{ri}-\varepsilon_{\mathrm{a}})^2}{n}}}{\varepsilon_{\mathrm{a}}}\,, \tag{48.1}$$

where n is the number of periods, and ε_{ri} and ε_{a} are the actual and average demand for spare parts in period i, respectively. The four resulting demand categories are represented graphically in Fig. 48.1.

The categorization scheme suggests different ways of treating the resulting categories according to the following characteristics:

- The condition ADI $\leq x$; $\mathrm{CV}^2 \leq y$ tests for stock-keeping units (SKUs), which are not very intermittent and erratic (i. e. faster moving parts, or parts whose demand pattern does not raise any significant forecasting or inventory control difficulties);
- The condition ADI $> x$; $\mathrm{CV}^2 \leq y$ tests for low-demand patterns with constant, or more generally not highly variable, demand sizes (i. e. not very erratic);
- The condition ADI $> x$; $\mathrm{CV}^2 > y$ tests for items with lumpy demand. Lumpy demand may be defined as a demand with large differences between each period's requirements and with a large number of periods having zero requests;
- Finally, the condition ADI $\leq x$; $\mathrm{CV}^2 > y$ tests for items with erratic (irregular) demand with rather frequent demand occurrences (i. e. not very intermittent).

In all cases, x denotes the cutoff value (ADI = 1.32) for ADI, which measures the average number of time periods between two successive demands, and y denotes the corresponding cutoff value ($\mathrm{CV}^2 = 0.49$) for CV^2, which is equal to the square of the standard deviation of the requirements divided by the average requirement over a number of time periods.

Forecasting systems generally depend on the category of part used. Therefore it is important to have two factors in order to indicate deviation from expected val-

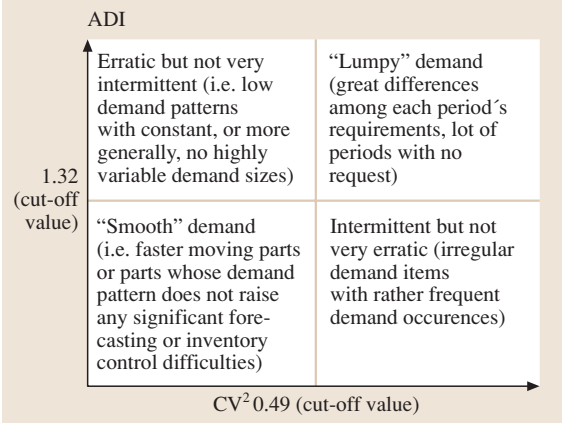


Fig. 48.1 Categorization of demand pattern

Table 48.1 A summary of selected forecasting methods

No.	Method	Abbreviation	Description
1	Additive Winter	AW	Assumes that seasonal effects are of constant size
2	Multiplicative Winter	MW	Assumes that seasonal effects are proportional in size to the local de-seasonalized mean level
3	Seasonal regression model	SRM	Used in time series for modelling data with seasonal effects
4	Component service life (replacement)	MTBR	Estimates of the service-life characteristics of the part (MTBR = mean time between replacement), derived from historical data
5	Croston	Croston	Forecasting in case of low and intermittent demand
6	Single-exponential smoothing	SES	Forecasting in case of low and intermittent demand
7	Exponential weighted moving average	EWMA, Holt	An effective forecasting tool for time series data that exhibits a linear trend
8	Trend-adjusted exponential smoothing	TAES	Forecasting time series data that have a linear trend
9	Weighted moving averages	WMA	A simple variation on the moving average technique that allows for such a weighing to be assigned to the data being averaged
10	Double-exponential smoothing	DES	Forecasting time series data that have a linear trend
11	Adaptive-response-rate single-exponential smoothing	ARRSES	Has an advantage over SES in that it allows the value of α to be modified in a controlled manner as changes in the pattern of data occur
12	Poisson model	POISSON	Models based on the Poisson distribution with the definition of a customer's service level
13	Binomial models	BM	Methods based on the binomial distribution

ues of demand with respect to both demand size and inter-demand interval. The performance of a forecasting method should vary with the level and type of lumpiness. A classification of research on intermittent demand forecasting can be arranged according to Willemain as follows:

1. extension of standard methods [48.2, 3] and variants of the Poisson model [48.4–10];
2. reliability theory and expert systems [48.11];
3. single exponential smoothing, Winter models [48.12–14],
4. Croston's variant of exponential smoothing [48.14–17];
5. bootstrap methods [48.18–21];
6. moving average and variants [48.22, 23];
7. models based on the binomial distribution [48.24–27].

The principle forecasting methods are briefly summarized in Table 48.1.

48.1.1 Exponential Smoothing

Exponential smoothing (ES) methods are widely used time-series methods when reasonably good forecasts are needed over the short term, using historical data to obtain a *smoothed* value for the series. This smoothed value is then extrapolated to become the forecast for the future value of the series. ES methods apply an unequal set of weights that decrease exponentially with time to past data; that is, the more recent the data value, the greater its weighting. In particular, the general form used in computing a forecast by the method of single-exponential smoothing (SES) is given by (48.2), where F_t represents the smoothed estimate, X_t the actual value at time t and α the smoothing constant, which has a value between 0 and 1. SES is best suited to data that exhibits a flat trend.

$$F_{t+1} = \alpha X_t + (1-\alpha) F_t \,. \tag{48.2}$$

When a trend exists, the forecasting technique must consider the trend as well as the series average; ignoring the trend will cause the forecast to underestimate or to overestimate actual demand, depending on whether there is an increasing or decreasing trend. In fact double-exponential smoothing (DES), which is useful when the historic data series are not stationary, applies SES twice and has the general

form:

$$F''_{t+1} = \alpha F_{t+1} + (1-\alpha) F_t \,. \tag{48.3}$$

48.1.2 Croston's Method

A little-known alternative to single-exponential smoothing is Croston's method, which forecasts separately the non-zero demand size and the inter-arrival time between successive demands using SES, with forecasts being updated only after demand occurrences. Let F_t and Y_t be the forecasts of the $(t+1)$th demand size and the inter-arrival time respectively, based on data up to demand t, and let Q_t be the inter-arrival time between two successive non-zero demand. Then Croston's method gives:

$$F_t = (1-\alpha) F_{t-1} + \alpha Y_t \,, \tag{48.4}$$

$$Y_t = (1-\alpha) Y_{t-1} + \alpha Q_t \,. \tag{48.5}$$

The predicted demand at time t is the ratio between F_t and Y_t

$$P_t = F_t / Y_t \,. \tag{48.6}$$

The SES and Croston methods are most frequently used for low and intermittent demand forecasting; in particular Croston's method can be useful with intermittent, erratic and slow-moving demand and its use is significantly superior to ES under intermittent demand conditions, according to the categorization scheme of Fig. 48.1. The straight Holt method, exponentially weighted moving average (EWMA), is also only applicable when there are low levels of lumpiness. The widespread use of the SES and mean time between replacement (MTBR) methods for parts with high variation (lumpy demand) are questionable as they consistently lead to poor forecasting performance, which remains poor as the demand variability increases. The only method that fits lumpy demand quite well is the weighted moving average (WMA) method and its superiority to ES methods has been proved: its use could provide tangible benefits to maintenance service organizations forecasting intermittent demand. By WMA we mean a moving average method in which, to compute the average of the most recent n data points, the more recent observations are typically given more weight than older observations.

48.1.3 Holt–Winter Models

Methods based on Winter's models [additive Winter (AW), multiplicative Winter (MW)] consider the seasonal factor and provide the biggest forecasting error when there is high variation (lumpy demand). While computing Holt–Winter filtering of a given time series, unknown parameters are determined by minimizing the squared prediction error; α, β and γ are the parameters of the Holt–Winter filter for the level, trend and seasonal components, respectively; if β is set to 0, the function will perform exponential smoothing, while if the γ parameter is set to 0, a non-seasonal model is fitted.

The additive Holt–Winter prediction function (for time series with period length p) is

$$\bar{Y}[t+h] = a[t] + h \cdot b[t] + s\,[t+1+(h-1)\,|p|] \tag{48.7}$$

where $a[t]$, $b[t]$ and $s[t]$ are given by

$$a[t] = \alpha(\bar{Y}[t] - s[t-p]) + (1-\alpha)(a[t-1] b[t-1]) \,, \tag{48.8}$$

$$b[t] = \beta(a[t] - a[t-1]) + (1-\beta) b[t-1] \,, \tag{48.9}$$

$$s[t] = \gamma(\bar{Y}[t] - a[t]) + (1-\gamma) s[t-p] \,. \tag{48.10}$$

The multiplicative Holt–Winter prediction function (for time series with period length p) is

$$\bar{Y}[t+h] = (a[t] + hb[t]) s[t+1+(h-1)|p|] \,, \tag{48.11}$$

where $a[t]$, $b[t]$ and $s[t]$ are given by

$$a[t] = \alpha \left(\frac{\bar{Y}[t]}{s[t-p]} \right) + (1-\alpha)(a[t-1] + b[t-1]) \,, \tag{48.12}$$

$$b[t] = \beta(a[t] - a[t-1]) + (1-\beta) b[t-1] \,, \tag{48.13}$$

$$s[t] = \gamma \left(\frac{\bar{Y}[t]}{a[t]} \right) + (1-\gamma) s[t-p] \,. \tag{48.14}$$

The function tries to find the optimal values of α and/or β and/or γ by minimizing the squared one-step prediction error, if they are omitted. For seasonal models starting values for a, b and s are detected by performing a simple decomposition in the trend and seasonal components using moving averages on the first period (a simple linear regression on the trend component is used for the starting level and trend). For level/trend models (no seasonal component) starting values for a and b are $X[2]$ and $X[2] - X[1]$, respectively. For level-only models (ordinary exponential smoothing), the starting value for a is $X[1]$.

48.2 Forecasting Methods

Table 48.2 Classification of forecasting methods, corresponding testing ground and applications

Forecasting methods	Testing ground							Main area of business application																							
																Sales aggregate			Sales disaggr.												
	Statistics	Operation research	Economics	Psychology	Long-range planning	Practice	Marketing	Production planning	Production scheduling	Inventories	Material requirements	Personnel scheduling	Personnel planning	Short term	Medium term	Long term	Short term	Medium term	Long term	Pricing	Advertising and promotion	Yearly budgeting	New products	R&D projects	Capital budgeting	Competitive analysis	Strategy
Judgmental																															
New product forecasting	•						•							•	•	•				•	•		•	•	•	•	•
Individual				•		•		•				•		•	•	•				•	•	•	•	•	•	•	•
Group																											
Decision rules				•					•	•	•	•															
Sales force estimates						•	•								•		•										
Juries of executive opinion						•	•								•			•				•					
Role playing				•														•									
Aggregate																											
Anticipatory surveys						•	•								•		•			•		•					
Market research							•											•		•	•		•	•			
Pilot programs, pre-market tests							•											•	•	•	•		•				
Quantitative																											
Naive						•		•	•	•		•		•			•			•							
Smoothing		•				•		•		•	•	•	•				•			•							
Decomposition			•					•	•			•	•		•			•				•					
Autoregressive moving average	•							•	•			•	•	•													
Explanatory																											
Vector autoregression	•		•											•	•					•	•	•	•				
Regression	•		•											•	•					•	•	•	•				
Econometrics	•		•											•	•					•	•	•	•				
Monitoring approaches	•	•				•			•	•	•	•					•										

Table 48.2 (cont.)

Forecasting methods	Testing ground: Statistics	Testing ground: Operation research	Testing ground: Economics	Testing ground: Psychology	Testing ground: Long-range planning	Testing ground: Practice	Testing ground: Marketing	Main area of business application: Production planning	Production scheduling	Inventories	Material requirements	Personnel scheduling	Personnel planning	Sales aggregate: Short term	Sales aggregate: Medium term	Sales aggregate: Long term	Sales disaggr.: Short term	Sales disaggr.: Medium term	Sales disaggr.: Long term	Pricing	Advertising and promotion	Yearly budgeting	New products	R&D projects	Capital budgeting	Competitive analysis	Strategy
Technological																											
Extrapolative																											
Growth curves	•				•														•					•	•	•	•
Time-independent comparisons						•													•					•	•	•	•
Historical and other analogies						•													•					•	•	•	•
Expert-based																											
Delphi					•													•					•	•	•	•	
Futurists					•													•					•	•	•	•	
Cross-impact matrices					•													•					•	•	•	•	

In our opinion, although many different classification schemes could be used, the most significant classification divides the major approaches to forecasting into three main categories, as summarized in Table 48.2 [48.28]: judgmental, quantitative and technological. Each category includes several types of methods, many individual techniques and variations. *Judgmental methods* are most commonly used in business and government organizations. Such forecasts are most often made as individual judgments or by committee agreements. Nevertheless quantitative methods are better than judgmental ones in determining spare-part inventory levels and we suggest judgmental methods only in extremis.

The second category – *quantitative methods* – is the type on which the majority of the forecasting literature has focused. There are three subcategories of these methods. *Time-series methods* seek to identify historical patterns (using a time reference) and then forecast using a time-based extrapolation of those patterns. *Explanatory methods* seek to identify the relationships that led to observed outcomes in the past and then forecast by applying those relationships to the future. *Monitoring methods*, which are not yet in widespread use, seek to identify changes in patterns and relationships; they are used primarily to indicate when extrapolation of past patterns or relationship is not appropriate.

The third category – *technological methods* – address long-term issues of a technological, societal, economic or political nature. The four subcategories here are extrapolative (using historical patterns and relationships as a basis for forecasts), analogy-based (using historical and other analogies to make forecasts), expert-based and normative-based (using objectives, goals and desired outcomes as a basis for forecasting, thereby influencing future events).

48.2.1 Characterizing Forecasting Methods

In describing forecasting methods there are seven important factors, which reflect their inherent capabilities and adaptability.

1. *Time horizon* – two aspects of the time horizon relate to individual forecasting methods. First is the span of time in the future for which different forecasting methods are best suited. Generally speaking, qualitative methods of forecasting are used more for longer-term forecasts, whereas quantitative methods are used more for intermediate- and shorter-term situations. The second impor-

tant aspect of the time horizon is the number of periods for which a forecast is desired. Some techniques are appropriate for forecasting only one or two periods in advance; others can be used for several periods. There are also approaches for combining forecasting horizons of different lengths.

2. *Pattern of the data* – underlying the majority of forecasting methods is an assumption about the type of pattern(s) found in the data to be forecast: for example, some data series may contain seasonal as well as trend patterns; others may consist simply of an average value with random fluctuations and still others might be cyclical. Because different forecasting methods vary in their ability to predict different types of patterns, it is important to match the presumed pattern(s) in the data with the appropriate technique.
3. *Cost* – generally three direct elements of costs are involved in the application of a forecasting procedure: development, data preparation and operation. The variation in cost obviously affects the attractiveness of different methods for different situations.
4. *Accuracy* – closely related to the level of detail required in a forecast is the desired accuracy. For some decision situations, plus or minus ±10% may be sufficient, whilst in others a small variation of 2% could spell disaster.
5. *Intuitive appeal, simplicity, ease application* – a general principle in the application of scientific methods to management is that only methods that are deeply understood are used by decision makers over time. This is particularly true in the area of forecasting.
6. *Number of data points required from past history* – some methods produce good results without consistent data from the past, because they are less affected by estimation errors in their input parameters.
7. *Availability of computer software* – it is seldom possible to apply a given quantitative forecasting method unless an appropriate computer program exists. Such software must be user-friendly and well conceived.

48.3 The Applicability of Forecasting Methods to Spare-Parts Demands

Companies have to select in advance an appropriate forecasting method matching their cyclical demand for parts. Particular attention has to be paid to the demand for service-part inventories, which is generally irregular and difficult to predict. A summary of the better forecasting methods, related to the categorization scheme in Fig. 48.1, is presented in Table 48.3.

Table 48.3 Summary of the better forecasting methods

Forecasting methods	Categorization of the demand			
	Intermittent	Erratic	Slow moving	Lumpy
Additive Winter (AW)		•	•	
Multiplicative Winter (MW)		•	•	
Seasonal regression model (SRM)		•	•	
Component service life (replacement)		•	•	
Croston	•	•	•	
Single-exponential smoothing (SES)		•	•	
Double-exponential smoothing (DES)		•	•	
Exponentially weighted moving average (EWMA)		•	•	
Trend-adjusted exponential smoothing		•	•	
Weighted moving averages	•	•	•	•
Adaptive-response-rate single-exponential smoothing		•	•	
Poisson models	•		•	
Binomial models	•	•	•	•

48.4 Prediction of Aircraft Spare Parts: A Case Study

The technical divisions of airlines are based on total hours flown and on the fleet size. With this data, the purchasing department tries to determine the quantity of stock necessary for a particular operating period. Alternatively, when new types of aircrafts are introduced, the airframe and engine manufacturers normally provide a recommended spares provisioning list, which is based on the projected annual flying hours, and includes forecast usage information for new aircraft. Original equipment manufacturers also provide overhaul manuals for aircraft components that support the assessment of required parts based on reliability information, i. e., on the specified components' operational and life limits. Consequently the forecast of spare parts is practically based on past usage patterns and personal experience.

Before any consideration about lumpiness and aircraft spare-parts forecast a discussion on the selection of the main variables used as the *clock* for spare-parts life evaluation is absolutely necessary. According to Campbell's study on maintenance records of the United State Air Force the demand for spare parts appears to be strongly related to flying hours; but this sometimes does not appear to be the best indicator, e.g., to forecast demand for landing gear, what matters is not how long the aircraft is in the air, but how often it lands, or radar components that work only when the aircraft is on the ground. In conclusion often flying hours are the best clock, but a demonstration of its effectiveness is necessary for each item.

In this study different forecasting methods have been considered; briefly:

1. Additive/Multiplicative Winter (AW/MW) For each forecast the optimal combination of level, trend and seasonal parameters is realized. Available values for each variable (level and trend) are 0.2 and 0.01; the seasonal length used is 12 periods.
2. Seasonal regression model (SRM). A multiplicative model with trend and seasonal components. The seasonal length is 12 periods.
3. Single-exponential smoothing (SES). The statistical software applied (Minitab 14.0©) supports the research of the optimal weight of the smoothing constant. The result is then the best forecast with this method.
4. Double-exponential smoothing (DES). Dynamic estimates are calculated for two components: level and trend; the software supports their optimization. In this case the best forecast with DES is also guaranteed.
5. Moving average (on the generic i-period) [MA(i)]. Moving averages (MAs) are calculated with different

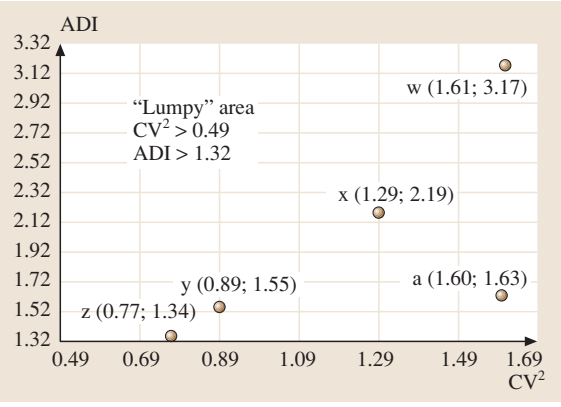


Fig. 48.2 CV^2 and ADI on monthly period for give representative *lumpy* items

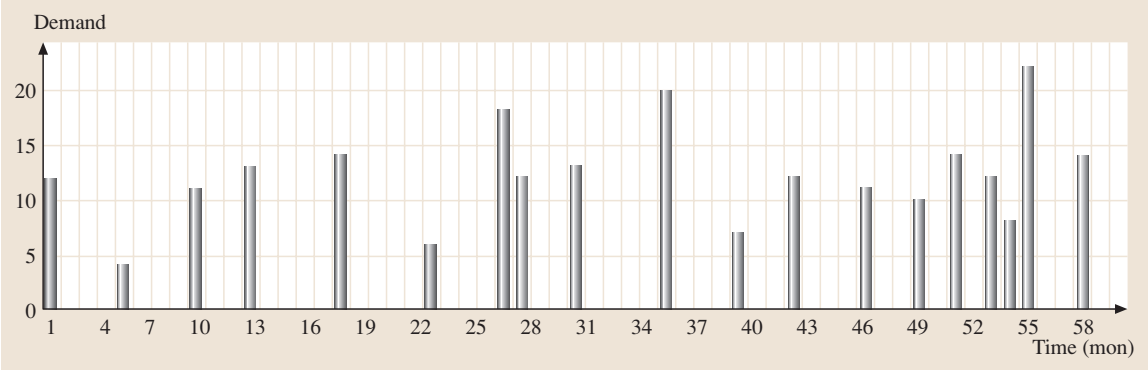


Fig. 48.3 Demand pattern for item z

Table 48.4 Comparison among some methods

Item z	MW	AW	SES	DES	MA(3)	MA(4)	MA(5)	MA(8)	SRM	EWMA (3)	EWMA (4)	EWMA (5)	EWMA (8)
MAD	4.04	3.71	4.54	5.74	5.16	4.80	4.97	4.43	3.53	3.92	3.88	4.06	4.01
MAD/A	0.58	0.53	0.65	0.82	0.74	0.68	0.71	0.63	0.50	0.56	0.55	0.58	0.57

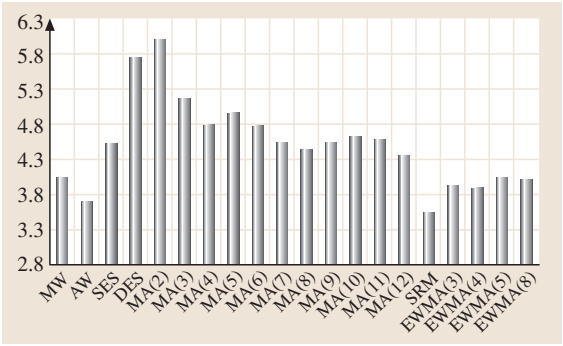


Fig. 48.4 MAD for item z

time horizons (i-period). The notation is MA(i). This analysis employs every period length from 2 to 12.

6. Exponentially weighted moving average [EWMA(i)]. In this case a weight optimization of smoothing coefficient for an MA series has also been realized. EWMA(i) is calculated for $i = 3, 4, 5$ and 8 periods.

Despite their importance in the literature [48.29–31], we do not evaluate and compare methods based on the Poisson approach because they are revealed as inadequate for the prediction of intermittent demand.

The case study deals with more than 3000 different items, with different levels of lumpiness: the Airbus fleet belonging to the Italian national-flag airline. For each component records relate to the daily demand level grouped in monthly interval of item usage, from 1998–2003. In terms of lumpiness these avionic spare parts are classified into five different classes of behavior; for each class a representative item, named a, x, y, z, w for confidentiality, is indicated.

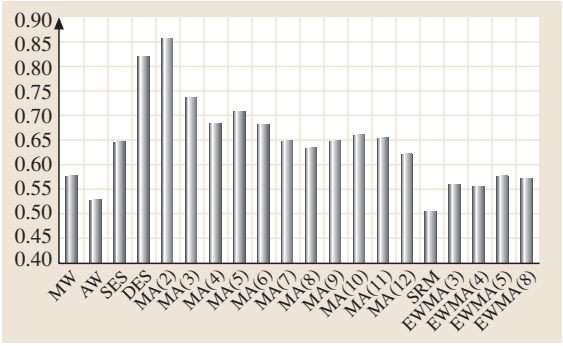


Fig. 48.5 MAD/A for item z

Figure 48.3 presents an exemplifying demand of item z. The five lumpiness levels are reported in Fig. 48.2.

The mean absolute deviation (MAD) of the forecast error is adopted as a performance index

$$\mathrm{MAD} = \frac{\sum_{i=1}^{n} |\varepsilon_{ri} - \varepsilon_{fi}|}{n} , \qquad (48.15)$$

where ε_{fi} is the forecasted demand for period i. Some authors propose the mean absolute percentage error (MAPE) for this comparison, but under lumpy conditions many item demands are zero, and as a consequence MAPE is not defined. For this reason some authors propose a similar index, called MAD/A, also defined when the demand for items is zero:

$$\mathrm{MAD/A} = \frac{\mathrm{MAD}}{\mathrm{AVERAGE}} , \qquad (48.16)$$

where AVERAGE is the average value of the historical demand for the item. The tracking signal (TS), as defined by *Brown* [48.32], is used to check if forecasts are in control or not.

$$\mathrm{TS}_t = \left| \frac{\mathrm{CUSUM}_t}{\mathrm{EMAD}_t} \right| , \qquad (48.17)$$

where $\mathrm{CUSUM}_t = (\varepsilon_{rt} - \varepsilon_{ft}) + \mathrm{CUSUM}_{t-1}$ and $\mathrm{EMAD}_t = \alpha\,|\varepsilon_{rt} - \varepsilon_{ft}| + (1-\alpha)\mathrm{EMAD}_{t-1}$.

Limit values of TS and the optimal α value (0.25) are derived from the approach of Alstrom and Madsen [48.33]. For the items analyzed forecasts are in control from the third year (i. e., their tracking signals respect the limits). The different forecasting methods are compared for all items and in particular for the proposed five components.

Table 48.4 and Figs. 48.4 and 48.5 show, respectively, some brief and full results of MAD and MAD/A for item z. Table 48.5 presents, for each representative item, the list of forecasting methods ordered

Table 48.5 Ranking based on performance evaluation (MAD)

Weight	*z*	*y*	*x*	*a*	*w*	Method	Total score	Average score
20	SRM	EWMA(3)	SRM	EWMA(4)	SRM	SRM	93	18.6
19	AW	SRM	AW	EWMA(3)	EWMA(5)	EWMA4	89	17.8
18	EWMA(4)	SES	MW	EWMA(5)	EWMA(4)	EWMA3	86	17.2
17	EWMA(3)	EWMA(4)	EWMA(5)	EWMA(8)	EWMA(3)	EWMA5	84	16.8
16	EWMA(8)	EWMA(5)	EWMA(4)	SES	SES	EWMA8	76	15.2
15	MW	EWMA(8)	SES	MW	AW	MW	60	15
14	EWMA(5)	AW	EWMA(8)	SRM	EWMA(8)	SES	75	15
13	MA(12)	MA(10)	EWMA(3)	MA(7)	MA(5)	AW	74	14.8
12	MA(8)	MW	MA(10)	MA(8)	MA(4)	MA12	51	10.2
11	MA(7)	MA(11)	MA(11)	MA(11)	MA(12)	MA11	49	9.8
10	SES	MA(12)	MA(9)	MA(4)	MA(9)	MA9	47	9.4
8	MA(11)	MA(6)	MA(8)	MA(12)	MA(11)	MA7	44	8.8
7	MA(10)	MA(8)	MA(7)	AW	MA(7)	MA8	43	8.6
6	MA(6)	MA(7)	MA(5)	MA(10)	MA(8)	MA4	35	7
5	MA(4)	MA(5)	MA(6)	MA(6)	MA(6)	MA5	32	6.4
4	MA(5)	MA(4)	MA(4)	MA(5)	MA(3)	MA6	29	5.8
3	MA(3)	MA(3)	MA(3)	MA(3)	DES	MA3	16	3.2
2	DES	DES	MA(2)	DES	MA(2)	DES	10	2
1	MA(2)	MA(2)	DES	MA(2)		MA2	7	1.4

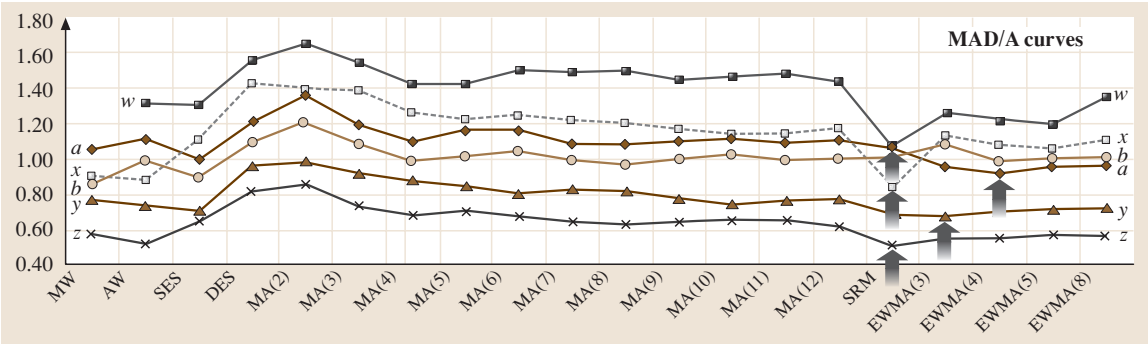


Fig. 48.6 MAD/A data and curves

by decreasing MAD, thus assigning a relative weight related to the ranking position: a simple elaboration of these weights permits a full comparison in terms of total and average scores (MW is not defined for item w, due to its characteristics).

By means of MAD/A it is possible to compare different forecasting methods for different items and their behavior in face of different lumpiness conditions; SRM, EWMA(i) and Winter are the best forecasting methods. This result is not related to the lumpiness level, at least for lumpiness represented by ADI < 3.3 and $CV^2 < 1.8$, which is the typical range for aircraft components. Some interesting observations can be drawn:

- Figure 48.6 clearly attests that item lumpiness is a dominant parameter, whilst the choice of the forecasting method is of secondary relevance; all methods for a slightly lumpy item (e.g. items y and z) generally perform better than the best method for a highly lumpy component (e.g. items x and w). However lumpiness is an independent variable and is not controllable;
- the average value of MAD/A, calculated for all forecasts generated by all methods, is 1.02. The aim of this study is to compare the different forecasting methods, but we can conclude that demand forecasting for lumpy items is very difficult and the results

are not very accurate. Moreover, lumpy demand is often equal to zero or one: all predictions lower than one must be rounded up to one. This phenomenon introduces another source of error;

- for a single component, the average fluctuation (in terms of MAD/A) of the ratio maximum/minimum, among different techniques, is about 1.55 (usually between 1.40 and 1.70); for a single forecasting method, the average fluctuation (in terms of MAD/A) of the ratio maximum/minimum, among different components, is about 2.17 (usually between 1.57 and 2.18). Thus, the demonstrating again the relevance of lumpiness.
- analyzing the effectiveness of a single model, research demonstrates (Tables 48.3 and 48.4) that the seasonal regression model (SRM), the exponentially weighted moving average [EWMA(i)] and the Winter model are the best forecasting methods. It is important to remember that the analyzed items are effectively representative of a population of aircraft spare parts. This result is not related to the lumpiness level, at least for lumpiness represented by ADI < 3.3 and $CV^2 < 1.8$ (the typical range for aircraft components).

In conclusion, intermittent demand for, usually highly priced, service parts is a very critical issue, especially for the prediction of lumpy demand, as is typical for avionic spare parts. In the literature forecasting for lumpy demand has not been investigated deeply, apart from Ghobbar's interesting research, and conflicting results are sometimes recovered. The introduction of the economic question is the final development: it is absolutely necessary to check the impact of stocking costs and out-of-stock components on the forecasting methods; an aircraft operator can incur costs of more than \$30 000 per hour if a plane is on the ground.

48.5 Poisson Models

For builders of high-technology products, such as automatic packaging machines, the supply of spare parts creates a strategic advantage with respect to their competitors, with particular regards to low-turnaround-index (LTI) parts conceived and designed for the satisfaction of a specific customer request. The strategic problem to solve is to determine the minimum number of spare parts required to avoid downtimes of the customer's plants for a specific period, called the covering period, which coincides with the time between two consecutive consignments.

The procedure actually used by a great number of manufacturers, called *recommended parts*, consists of the creation, at the design stage, of different groups of replaceable parts with different covering times for every functional machine group. This methodology is very qualitative and depends strongly on the opinion of the designer; moreover, it does not consider information feedback from customers, and usually overestimates the number of spare parts with respect to the real demands of customers. Even though this avoids plant downtimes, which are absolutely forbidden due to the high costs of production loss, it normally creates excessive and expensive stocks, with undesirably high risks of damage and obsolescence. For LTI items the usual economic batch or safety stock methods are not suitable to forecast the amount of spare parts required. For such a situation a lot of different approaches have been developed in recent years, usually based on the Poisson distribution; of these, conditioning of the stock level to minimal availability or to minimum total cost (Fig. 48.7) are considered the most interesting.

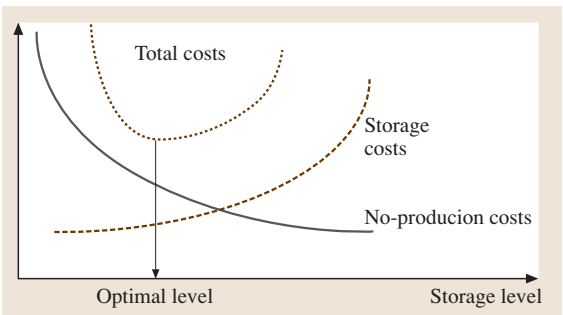


Fig. 48.7 Economic approach

Every study reported in the literature [48.34, 35] assumes that an item's failure time (for spare-parts demand) is exponentially distributed and, as a consequence, the failure rate $\lambda(t)$ is independent of time; this simplifying hypothesis is due to the difficulties in estimating real values of mean time before failure (MTBF). Finally it is important to underline that the quantity of spare parts and its temporal distribution also represent

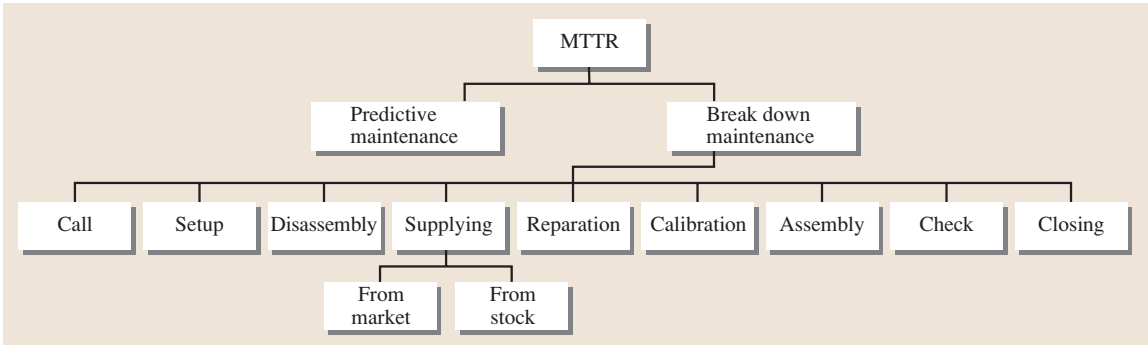


Fig. 48.8 MTTR structure

strategic information during negotiations with customers for the purchase of plants and the quantification of related costs.

48.5.1 Stock Level Conditioned to Minimal Availability

This method firstly needs to calculate the asymptotic availability A by the known formula:

$$A = \frac{\text{MTBF}}{\text{MTBF}+\text{MTTR}} \,. \tag{48.18}$$

The mean time to repair (MTTR) term is derived from different factors, as shown in Fig. 48.8.

Its value depends strongly on the spare part being on consignment, i. e. on hand, or not, and can be calculated by the formula

$$\text{MTTR} = T_1 + \int_0^{T_S} (T_S - T_x) f(T_x) \mathrm{d}T_x = \text{MTTR}(N) \,, \tag{48.19}$$

where T_1 is the amount of time due to factors except supply time (for instance, disassembling), T_S is the supply lead time for unavailable components, N is the number of spare parts available in stock at time zero, T_x is the time interval between the instant when the consumption of the part reaches the value N (empty stock situation) and the consignment of the spare part, and $f(t)$ is the failure density distribution.

It is worth noting that, for increasing N, we get decreasing MTTR, increasing availability A and the falling downtime costs. Secondly the method affords the quantitative definition of the storage cost, which requires the definition of the average number of parts stored during the time of supply T_S. If the warehouse contains N parts at time zero, the probability P_N of N failures in T_S can be described by the Poisson formula

$$P_N = \frac{(\lambda T_S)^N \cdot \mathrm{e}^{(-\lambda T_S)}}{N!} \,. \tag{48.20}$$

In the same way it is possible to calculate the probability of one, two, or N failures.

Let R indicate the cost of each spare part, and s be the stocking cost index per year; the annual stock costs C can be evaluated by the formula $C = Rs[NP_0 + (N-1)P_1 + (N-2)P_2 + \ldots P_{N-1}]$, which can be used in an iterative manner to find the optimum level N that leads to a minimum for the cost C, while allowing the minimum level of availability $A_{\min}(N)$ to guarantee on-time technical requests to be satisfied (for example, safety questions or productivity level)

$$\begin{cases} \min \left\{ C = Rs \left[NP_0 + (N-1)P_1 \right. \right. \\ \qquad \left. \left. + (N-2)P_2 + \cdots + P_{N-1} \right] \right\} \\ \text{subject to } A(N) = \frac{\text{MTBF}}{\text{MTBF}+\text{MTTR}(N)} \geq A_{\min} \,. \end{cases} \tag{48.21}$$

48.5.2 Stock Level Conditioned to Minimum Total Cost

The aim of this method is to determinate the total amount N of replaceable parts that minimizes the total cost function C_{tot} defined by

$$C_{\text{tot}}(N) = C_1 + C_2 \,. \tag{48.22}$$

The warehousing cost term C_1 can be estimated as in (48.21), while for the cost C_2 it is necessary to quantify the probability of stock-out situations. During the time T_S production losses could occur if the number of failures exceeds the number N of parts supplied at the consignment time, assumed to be zero. The cumulative

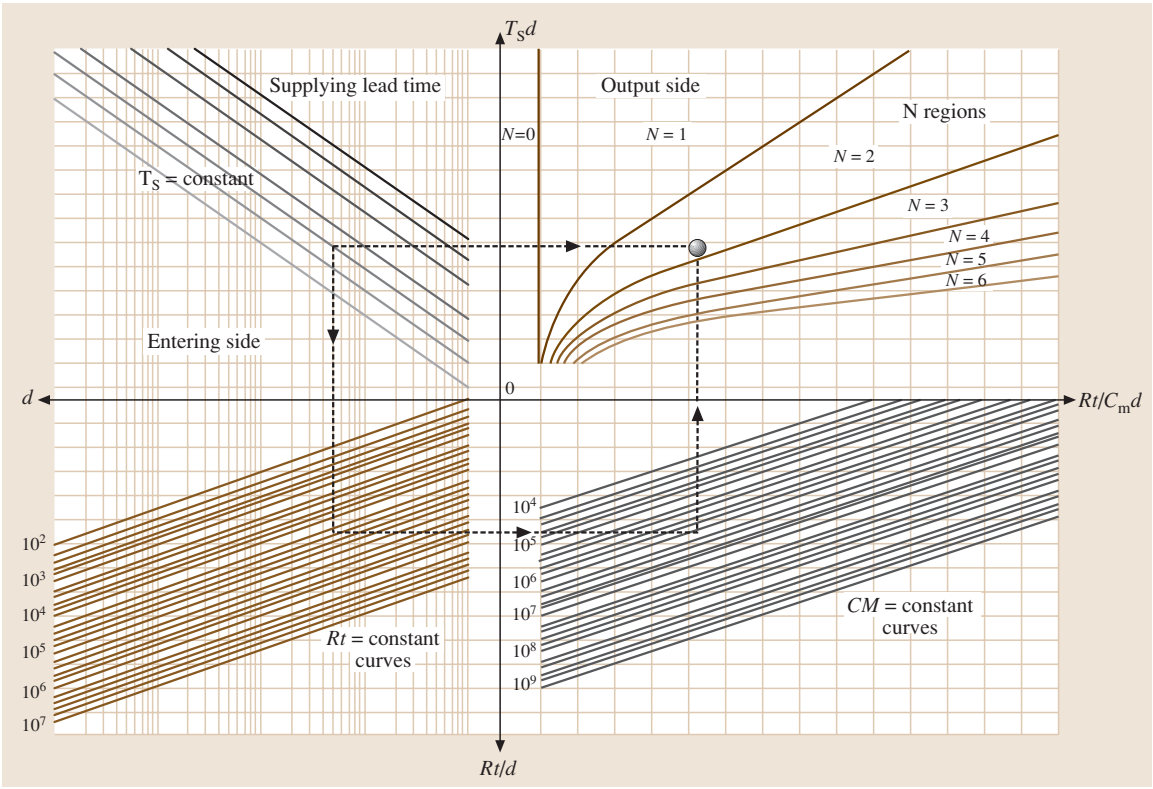


Fig. 48.9 Graphical abacus

probability, calculated by the Poisson distribution, is

$$P = P_{N+1} + P_{N+2} + P_{N+3} + \dots . \tag{48.23}$$

Let d indicate the annual part consumption of a customer and CM the cost corresponding to a loss of production; the term due to stock-out is

$$C_2 = CMdP . \tag{48.24}$$

For a rapid choice it is possible to employ a user-friendly abacus (Fig. 48.9).

48.6 Models Based on the Binomial Distribution

Industrial applications show that methodologies based on the Poisson formula usually overestimate the actual replacement consumption. To overcome this problem we present a new quantitative procedure that, in contrast to the Poisson methods, does not assume that requests for parts are linear over time.

The innovative approach calculates the requirement for components, for a given covering period T, by the addition of two addenda x_1 and x_2: the first is related to the wear damage of the replaceable component and can be deduced from the MTBF value, while the second refers to the randomness of breakdowns and covers the possibility of failures in advance of the average situation. The optimal number of replacements is $N = x_1 + x_2$.

Let n be the number of different employments of a component in several machines owned by the customer and let T be the covering period; x_1 can be expressed by:

$$x_1 = \text{int}\left(\frac{T}{\text{MTBF}}\right) n . \tag{48.25}$$

This average term assumes interesting values only in the presence of high consumption of the component, in par-

Table 48.6 Example of N evaluation for a specific item (code 0X931: pin for fork gear levers)

Input	
MTBF (h)	3945
Positions (n)	26
Confidence level	97%
Supplying time (h)	2300

Output	
Failure ratio λ	2.53×10^{-4}
X_1	0
Tresidual (h)	2300
P (Tresidual)	0.442
X_2	See table
$N = X_1 + X_2$	16

ticular in the rare situations when a LTI part has a lot of applications, indicated by n. Anyway this term x_1 represents scant information; we have to consider the second term, which corresponds to the number of parts needed to obtain the required value of the customer service level (LS) in the residual time T_{residual}, defined as the residue of the ratio between T and MTBF. The customer service level is the probability that the customer finds the parts during the remaining period, and can be fixed separately according to strategic and economic assessments.

The value of x_2 is obtained as follows

$$T_{\text{residual}} = T - \text{int}\left(\frac{T}{\text{MTBF}}\right)\text{MTBF}\,, \tag{48.26}$$

$$p = Q(T_{\text{residual}}) = 1 - \text{e}^{-\left(\frac{T_{\text{residual}}}{\text{MTBF}}\right)}\,, \tag{48.27}$$

where p represents the failure probability during T_{residual}. Using p and the binomial distribution it is easy to calculate the probability that a component (with n applications) requires fewer than x_2 replacements in T_{residual}:

$$P\left[x \le x_2; n; Q(T_{\text{residual}})\right] = \sum_{i=0}^{x_2}\binom{n}{i}(1-p)^{n-i}p^i\,. \tag{48.28}$$

As a consequence it is possible to quantify the confidence level for no stock-outs to compare with the customer satisfaction as

$$\text{LS}(x_2) = 1 - P\left[x \le x_2; n; Q(T_{\text{residual}})\right]\,. \tag{48.29}$$

The main innovative result is that the procedure, in contrast to other methods, does not consider the total requests for spare parts to be linear with time; it tries to set the best moment for supply in order to maximize the customer service level without increasing the average number of spare parts. In fact the new method respects the average consumption through the term x_1 and increases the levels of customer service by planning requirements for spare parts in the residual time through the term x_2.

48.6.1 An Industrial Application

This procedure is successfully running on PC systems in an Italian company that is a leader in manufacturing

Table 48.6 (cont.)

X_2 (items)	LS	$(X_1 + X_2)$ (items)	X_2 (items)	LS	$(X_1 + X_2)$ (items)
0	0.000	0	14	0.883	14
1	0.000	1	15	0.943	15
2	0.000	2	16	0.976	16
3	0.000	3	17	0.991	17
4	0.002	4	18	0.997	18
5	0.007	5	19	0.999	19
6	0.022	6	20	1.000	20
7	0.055	7	21	1.000	21
8	0.118	8	22	1.000	22
9	0.218	9	23	1.000	23
10	0.351	10	24	1.000	24
11	0.505	11	25	1.000	25
12	0.658	12	26	1.000	26
13	0.787	13			

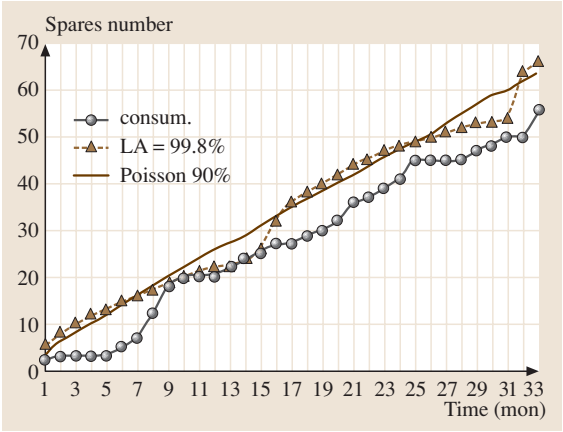


Fig. 48.10 Forecast and applications for the pin

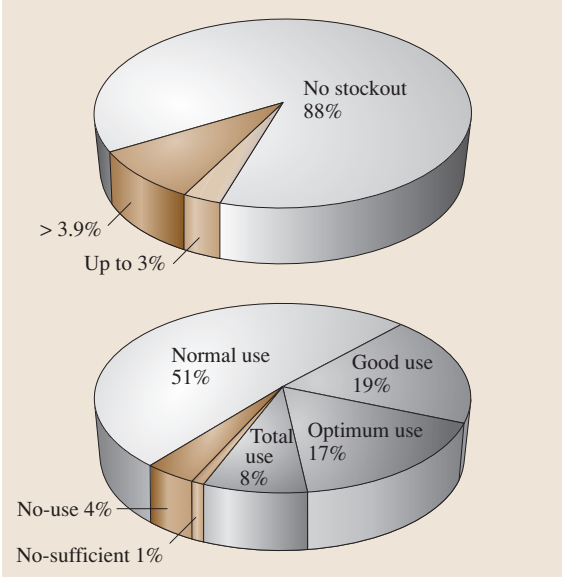


Fig. 48.11 Simulated stock-out periods (months) and real utilization of dispatched replacements

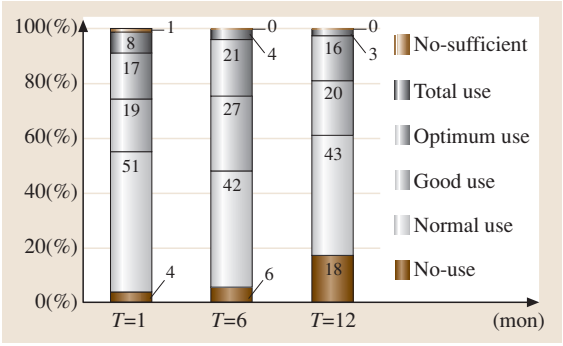


Fig. 48.12 Real use compared to supply time T

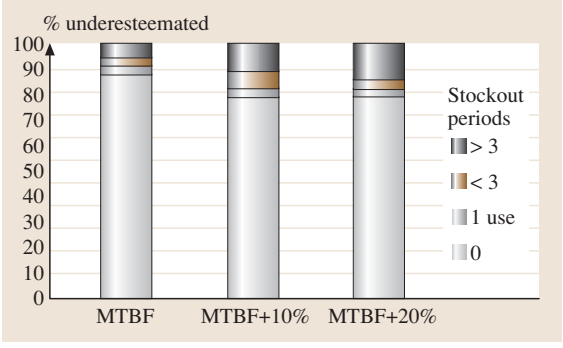


Fig. 48.13 Sensitivity to MTBF evaluation

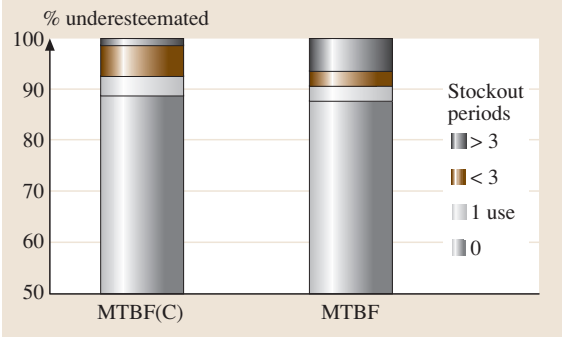


Fig. 48.14 MTBF correction effect on stock-out

of packaging machines. The supply of spare parts creates a strategic advantage over competitors, because the automatic packaging machines usually present a long life cycle and contain a lot of functional groups, often conceived and designed ad hoc. The economic impact of replacement activity is not negligible: it usually amounts to 15% of global business volume. A good forecast of spares parts can surely simplify manufacturer production planning. Before its industrial real-time application the innovative procedure was tested to forecast the consumption of 190 different spare parts indicated by several customers over a 33-month period.

The experimental results of these procedure were evaluated by two performance indexes:

1. percentage of spare parts without stock-out periods,
2. effective utilization of replacements by the customer.

As a first test the new procedure was applied with the restrictive hypothesis of a covering period of one

month: using an LS value of 99.8% the method performed well for 166 replaceable parts and there was very good correlation between predicted and actual customer consumption. This is shown in the comparison in Table 48.6 between actual consumption and the Poisson linear forecast (LS = 90%) for a pin for fork gear levers with 26 applications in the customer's machine park.

88% of the components investigated did not suffer from stock-out in any month; 3% presented less than three months of under-evaluation and 9% had more than three months in stock-out. On the other hand the utilization index showed that 87% of the components had normal, good or optimal customer use: in other words they did not remain in the spare-parts warehouse for more than 15 days before installation (Fig. 48.11).

Moreover, parts with bad forecasts were investigated to understand the reasons, with very encouraging results; errors were usually caused by preventive maintenance operations, machine revisions, changes of suppliers or changes in the application of the component not pointed out by the customer. The effect of the extension of the covering period was analyzed by testing the new method with different values of T. For increasing T we obtain an increasing stock of spare parts but the number of stock breakages strongly decreases: in fact for $T = 1$ month the method performed well for 88% of the parts investigated, and this percentage increased to 90% for $T = 3$ months, 95% for $T = 6$ months and to 98% for $T = 1$ year. Therefore it was possible to study the optimum extension of the covering period, that for the 190 components investigated was found to be equal to three periods (Fig. 48.12).

Some simulations with different values of MTBF show the influence of its approximate evaluation: values of MTBF overestimated by 10% and 20% reduce the performance of the method respectively by 6% and 9% in terms of the percentage of spare parts without stock-out periods (Fig. 48.13).

In spite of this important conclusion, it is important to remember that MTBF values have to be updated, starting from the initial value of $\mathrm{MTBF}_{\mathrm{initial}}$, by feedback information from the customers; the most suitable control parameter is the component quantity Y employed by the customer during the covering period T and the relation (48.30) that gave the best results, as shown in Figs. 48.14 and 48.15:

$$\mathrm{MTBF}_{\mathrm{updated}} = \frac{\mathrm{MTBF}_{\mathrm{initial}}\,(\mathrm{MTBF}_{\mathrm{initial}}) + \frac{nT}{Y}(nT)}{\mathrm{MTBF}_{\mathrm{initial}} + nT}\,, \tag{48.30}$$

where $\mathrm{MTBF}_{\mathrm{updated}}$ is the weighted average of $\mathrm{MTBF}_{\mathrm{initial}}$, and $\frac{nT}{Y}$ (weights in round brackets).

48.7 Extension of the Binomial Model Based on the Total Cost Function

The proposed model required the assumption of a specific spare-part LS defined as the probability of finding the part in case of breakdown. Some simulations with different values of LS show that it is important to assume LS $\leq$ 80% and to reserve LS $\geq$ 90% for particular situations, e.g. customers placed in a distant country or without skilled workers. It is possible to determine the number N of replaceable parts needed and therefore the LS value capable of minimizing a total cost function defined by the sum of costs due to storage and production losses.

48.7.1 Service-Level Optimization: Minimum Total Cost Method

The aim of this paragraph is to determine the requirement N of replaceable parts capable of minimizing a total cost function defined by the sum of production losses costs C_1 and storage costs C_2. During the time T_S production losses could occur if the number of failures exceeds the number N of supplied parts that are available after the consignment at time zero. The corresponding cumulative probability can be calculated by formula:

$$\begin{aligned} P &= P(N+1)+P(N+2)+P(N+3)+\ldots \\ &= 1-\mathrm{LS}(x_2) = 1-\mathrm{LS}(N-x_1)\,, \end{aligned} \tag{48.31}$$

$$\mathrm{LS}(N-x_1) = P[X \leq N-x_1,\ n,\ Q(T_{\mathrm{residual}})]\,. \tag{48.32}$$

If d and CM represent, respectively, the customer annual part consumption and the cost for a production lack, the total cost C_1 due to stock-out is

$$C_1 = CMdP\,. \tag{48.33}$$

The storage cost C_2 requires the definition of the average number of parts stored during the supplying time T_S. Two different situations are possible related to the spare-part MTBF and T_S:

Table 48.7 LS % and minimum cost related to $T_s d$ and $Rt/(C_m d)$– no. of employments $n = 5$

LS % ($n = 5$)	$T_s d$							
$Rt/(C_m d)$	5	4	2	1	0.8	0.2	0.02	0.02
9	0.7	1.8	13.5	37	44.9	81.9	98	99.8
3	6.5	13.1	13.5	37	44.9	81.9	98	99.8
1.5	26.4	40.6	46.8	37	44.9	81.9	98	99.8
0.9	60.5	40.6	46.8	78	44.9	81.9	98	99.8
0.3	89.9	74.3	79.6	78	83.9	81.9	98	99.8
0.1	100	94.9	95.7	95	97.4	81.9	98	99.8
0.03	100	100	99.6	100	97.4	98.6	98	99.8
0.003	100	100	100	100	99.8	99.9	100	99.8
0.0003	100	100	100	100	100	100	100	100

Case a. MTBF $< T_S$; in this case $N = x_1 + x_2 = x_2$ because $x_1 = 0$.

If the warehouse contains N parts at time zero, the probability P_N of N failures in T_S can be calculated using (48.21). Let R indicates the cost of each spare part, and t the annual stocking cost index; the global annual storage cost C can be evaluated by

$$C = Rt \sum_{k=0}^{N} \left[\binom{N}{k} (1-p)^{N-k} p^k \right] (N-k) \,. \quad (48.34)$$

Case b. MTBF $\geq T_S$

In this case the definition of the average number of parts stored during the supply time T_S has to take in account both contributions, in terms of average stock $S(x_1)$ and $S(x_2)$, of x_1 and x_2. LS is connected to x_2, and the minimum real value of N is therefore x_1. We can get the contributions in (48.36) by the previously defined values of x_1 and x_2. The annual stock cost C can be evaluated by

$$C = Rt\,[S(x_1) + S(x_2)] \,, \quad (48.35)$$

which can be used in an iterative process to find the optimum level N according to the minimization of previous cost C.

$$S(x_1) = \frac{1}{2} \left(\frac{x_1 \mathrm{MTBF}}{T_S} \right) ;$$

$$S(x_2) = \frac{x_2 \mathrm{MTBF}}{T_S} + \frac{\sum_{k=0}^{N-x_1} \binom{N-x_1}{k} (1-p)^{N-x_1-k} p^k}{T_S} \times (N - x_1 - k)\,(T_S - \mathrm{MTBF}) \,. \quad (48.36)$$

48.7.2 Simulation and Results

A simulation model has been designed in order to find the optimum value of LS for different values of the parameters. Input data are the MTBF, number of employments n, time for supply T_S, cost of each spare part R, annual stocking cost index t, downtime cost per hour C_m, MTTR, total hours per year of uptimes plus

Table 48.8 LS % and minimum cost related to $T_s d$ and $Rt/(C_m d)$– no. of employments $n = 15$

LS % ($n = 5$)	$T_s d$							
$Rt/(C_m d)$	5	4	2	1	0.8	0.2	0.02	0.02
9	4.7	10.2	13.5	37	45.0	81.9	98	99.8
3	34.8	28.2	42.5	75	45.0	81.9	98	99.8
1.5	57.3	52	71.4	75	81.9	81.9	98	99.8
0.9	76.9	73.8	71.4	75	81.9	81.9	98	99.8
0.3	89.8	88.5	89.3	93	96	98.4	98	99.8
0.1	96.4	95.9	96.9	99	99.4	98.4	98	99.8
0.03	99	98.8	99.3	100	99.4	99.9	98	99.8
0.003	100	100	100	100	99.9	99.9	100	100
0.0003	100	100	100	100	100	100	100	100

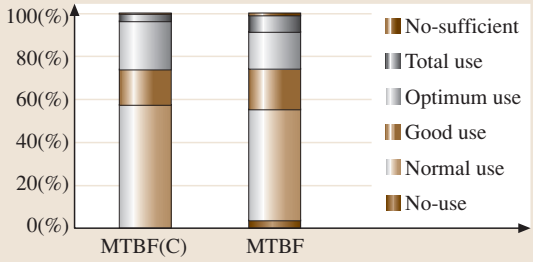


Fig. 48.15 MTBF correction effect on items' utilization

downtimes H, the customer annual part consumption d. Assuming for instance MTBF = 10 000 h and $n = 5$, the optimum value of LS versus the two variables $(Rt)/(C_m d)$ and $T_S d$ is reported in Tables 48.7 and 48.8 (for $n = 5$ and $n = 15$, respectively). It is worth stating that LS must be close to 100% when $C_m \gg R$, while in the opposite case the optimum LS is a function of $T_S d$, and always tends upwards for decreasing n.

Figure 48.16 shows how to employ the abacus.

Table 48.9 Optimization of T_s for fixed number of spare parts N

Component	$N = x_1 + x_2$	$T_s(d)$	LS(%)
Support grid	3	400	99
Clamp	4	90	98
Special gasket	5	90	95

48.7.3 An Industrial Application

This case study is related to an important producer of steam boiler systems that actually manufacturing components for internal use and for replacement ordered by customers according to a fixed economic order quantity (EOQ). The application deals with the optimization of the supply time in order to reduce the total management costs of spare parts at the assigned EOQ [48.36]; that is, the aim is to define the time between consignments capable of reducing total costs for the same value of EOQ. Three components (a support grid, the clamp and a special gasket) are considered, with a downtime calculated

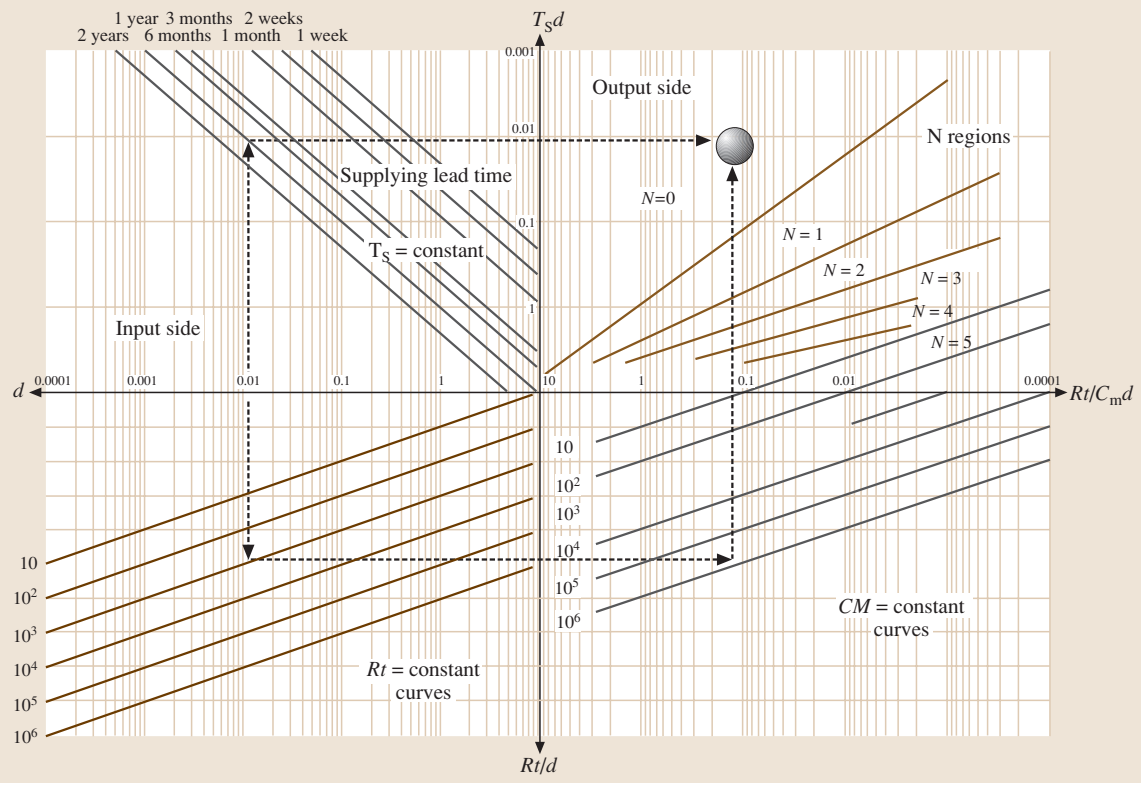


Fig. 48.16 Graphical solution for this methodology

as

$$\text{Downtime} = \text{MTTR} \times dn \,. \tag{48.37}$$

In particular the support grid has $n = 5$ employments, $H = 1760$ total hours per year, $d = 0.5$ annual part consumption [unit/year] on the whole, MTTR = 10 h, MTBF = 8800 h downtime cost per hour $C_m = 1000\,€/h$ and, as a consequence, $CM = C_m\text{MTTR} = 10\,000\,€$ cost for a production stock-out. This component was supplied in fixed EOQ with $N = 3$ elements, at a cost per unit of $R = 100\,€/\text{unit}$ ($R \ll C_m$); the relative risk of damage and obsolescence suggests $t = 0.1$ for the annual stocking cost index. Entering the abacus with these values for N, CM and Rt (Fig. 48.15) we obtain as a result the optimal time for supply $T_S = 400$ d; by means of (48.31) and (48.32) with $N = x_1 + x_2 = 0 + 3$ ($x_1 = 0$ because MTBF $< T_S$) we determine LS $\approx 99\%$; this very high level is due to the low values of t and the rate R/C_m.

The results following the use of the graphical abacus are shown in Table 48.9 for the whole set of components.

The results are summarized in Fig. 48.17 and compared with the output of an existent minimum-cost method based on the Poisson distribution.

48.8 Weibull Extension

This innovative methodology can be extended to the whole lifetime by implementing the Weibull failure-rate function to the stock level of LTI spare parts level in maintenance systems with declared wear conditions. The Weibull distribution is one of the most commonly used lifetime distributions and is flexible in modeling failure-time data, as the corresponding failure-rate function can vary or be assumed to be constant. The literature offers a lot of papers dealing with models for bath-tub-shaped failure rates. For example *Hjorth* [48.37] proposed a three-parameter distribution; *Mudholkar* and *Srivastava* [48.38] introduced an exponential Weibull distribution; *Chen* [48.39] spoke about a two-parameter lifetime distribution with a bath-tub shape or an increasing failure-rate function; *Xie* [48.40] wrote a very interesting paper about a model that can be seen as a generalization of the Weibull distribution and tries to improve the procedure for estimation of the parameters.

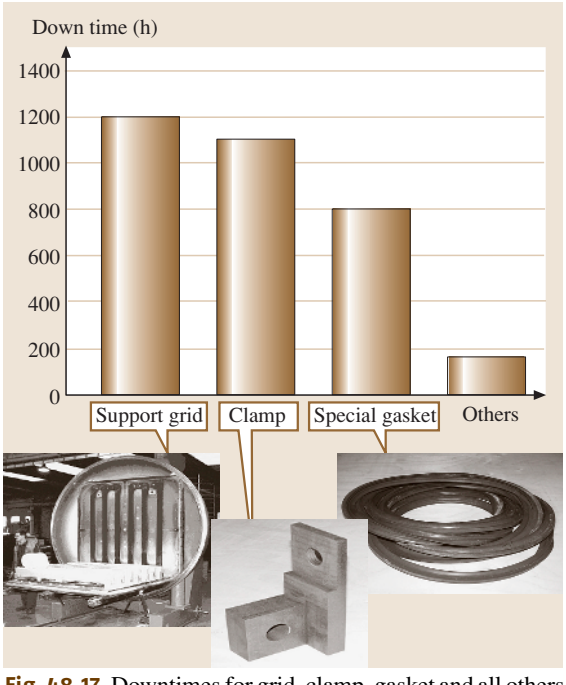


Fig. 48.17 Downtimes for grid, clamp, gasket and all others components (cumulative)

Estimation of the well-known parameters η (scale) and β (shape) in a Weibull distribution can be performed graphically but this is not accurate unless there is a large sample size, which is not always the case for LTI spare parts; anyway we focus our attention to the final zone in the traditional *bath-tub* wear model, and our aim is to understand whether the hypothesis of constant failure rate in our previous works increases the spare-parts costs, in comparison with more sophisticated distributions. For this reason we developed our model using the traditional Weibull distribution, but this could be extended to any of the models mentioned above.

48.8.1 The Extension of the Modified Model Using the Weibull Distribution

Using historical data it is possible to determine the cumulative percentage of component failures related to their lifetime. The graphical approach of Fig. 48.18 permits the definition of the Weibull distribution parameters η (scale parameter) and β (shape parameter). This is possible by Plait transformation (Weibull transformation); starting from the failure rate calculated as in (48.38),

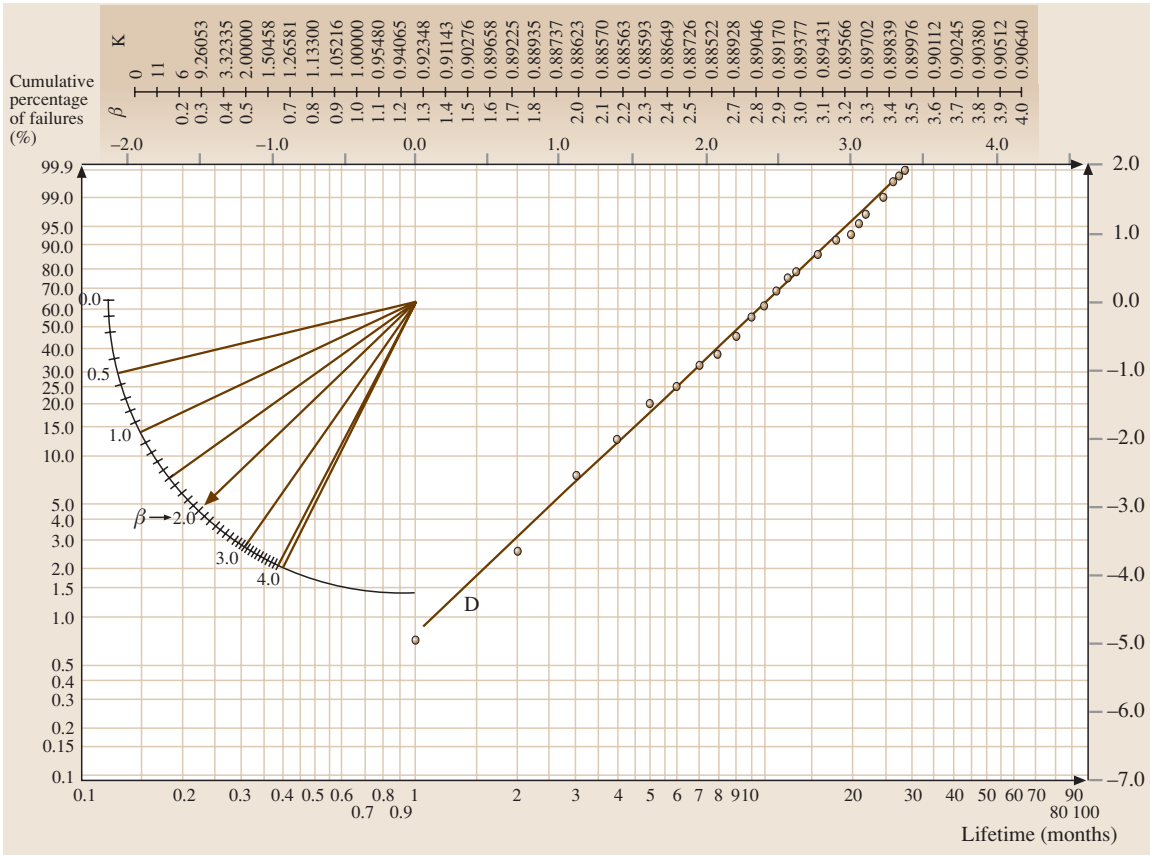


Fig. 48.18 Graphical estimation of the β value

the reliability in (48.39), the cumulative distribution functions in (48.40), and the definition of a normalized parameter x using (48.41), find a linear correlation between the parameter x and the cumulative distribution function (48.42), represented in Fig. 48.18

$$\lambda(t) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{t}{\eta}\right)^{\beta} = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} = \frac{\beta}{\eta^{\beta}} t^{\beta-1} , \tag{48.38}$$

$$R(t) = \mathrm{e}^{-\left(\frac{t}{\eta}\right)\beta} , \tag{48.39}$$

$$F(t) = 1 - R(t) = \mathrm{e}^{\left(-\frac{t}{\eta}\right)^{\beta}} , \tag{48.40}$$

$$x = -\frac{t}{\eta} , \tag{48.41}$$

$$\beta \ln(x) = \ln\left[\ln\left(\frac{1}{1-F(x)}\right)\right] . \tag{48.42}$$

The optimal replacement number of LTI spare parts is also given by relation (48.25), whilst the relations (48.27) and (48.28) are modified by the Weibull parameters η and β:

$$p = Q\left(T_{\text{residual}}\right) = 1 - \mathrm{e}^{-\left(\frac{t}{\eta}\right)^{\beta}} , \tag{48.43}$$

$$P\left[x \le x_2; n; Q\left(T_{\text{residual}}\right)\right] = \sum_{i=0}^{x_2} \binom{n}{i} (1-p)^{n-i} p^i . \tag{48.44}$$

It is possible to quantify the no-stock-out confidence level to compare with customer satisfaction as $\mathrm{LS}(x_2) = 1 - P[x \le x_2; n; Q(T_{\text{residual}})]$. As previously stated LS must be close to 100% when $C_{\mathrm{m}} \gg R$, while in the opposite case the optimum LS is a function of $T_{\mathrm{S}}d$; in this case, for fixed MTBF, the optimum LS increases with the number of employments n.

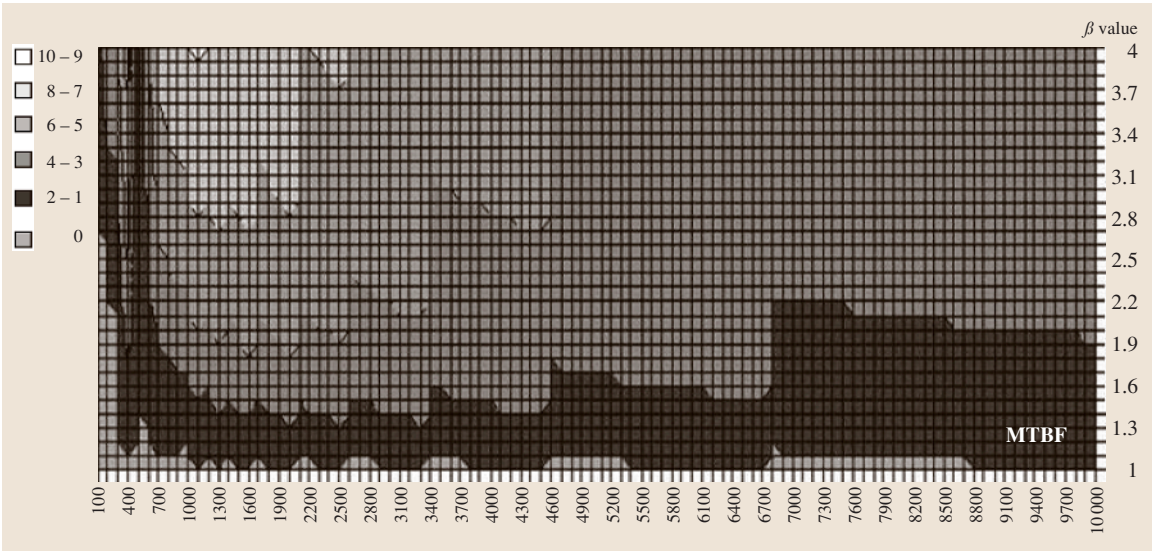


Fig. 48.19 Difference between optimal numbers of replacements calculated by (48.29) or (48.44) with respect to the average life of the component MTBF and β values

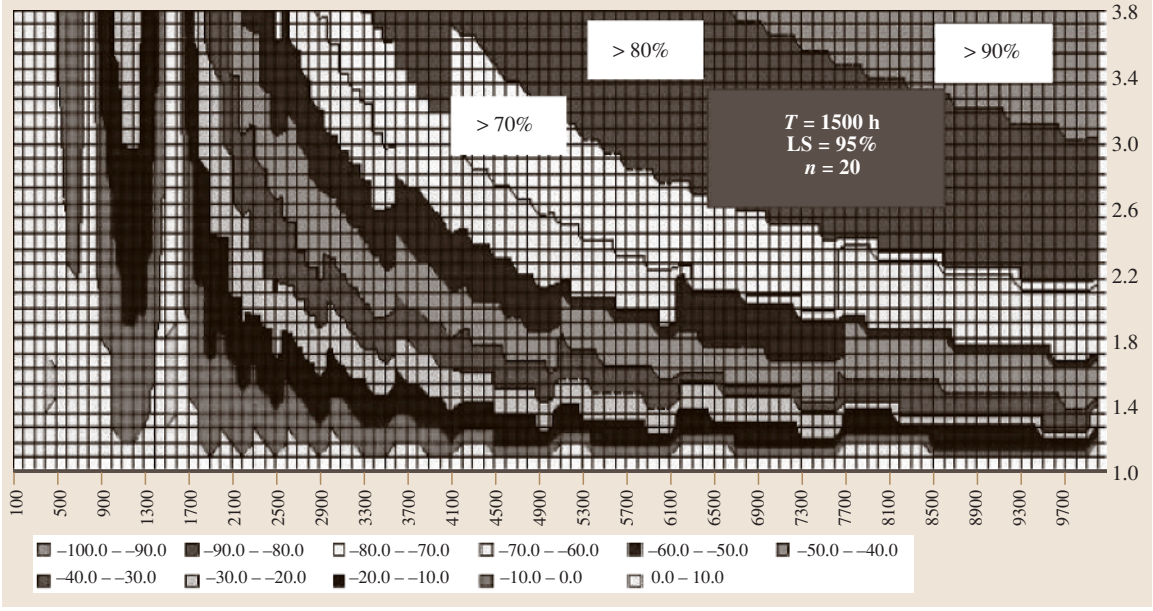


Fig. 48.20 Number of spare parts saved

48.8.2 Simulation and Results

This extended model is compared with previously proposed models for different values of the parameters involved. The relation between the parameters MTBF, T_S and β appears very interesting. In fact the first two parameters are fundamental to finding the quantity x_2, see (48.29) and (48.44), while β indicates the gap from the hypothesis of constant failure rate. The surface of Fig. 48.19 (with T_S equal to 500 h and a customer service level of 95%) relates the difference between optimal replacement numbers calculated

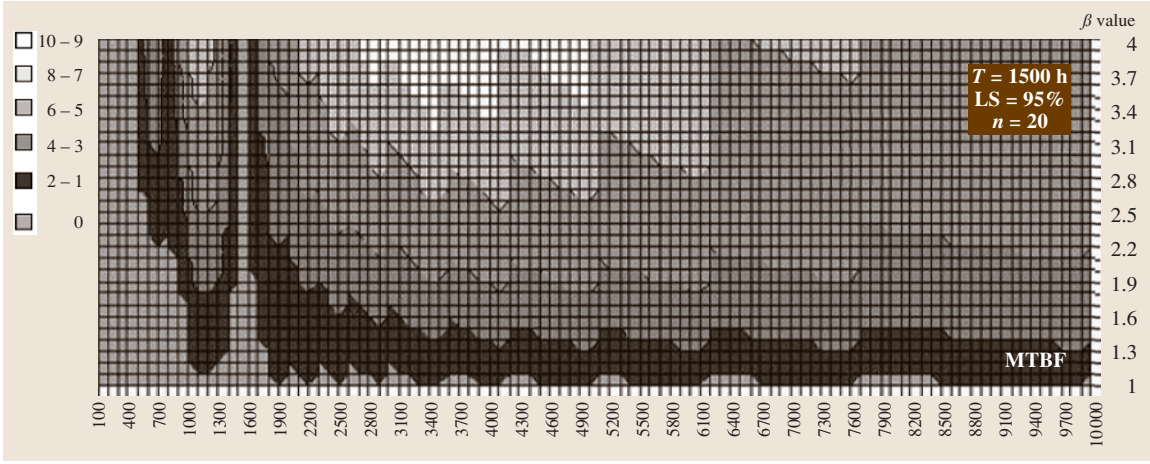


Fig. 48.21 Share of saving with respect to hypothesis of constant failure rate

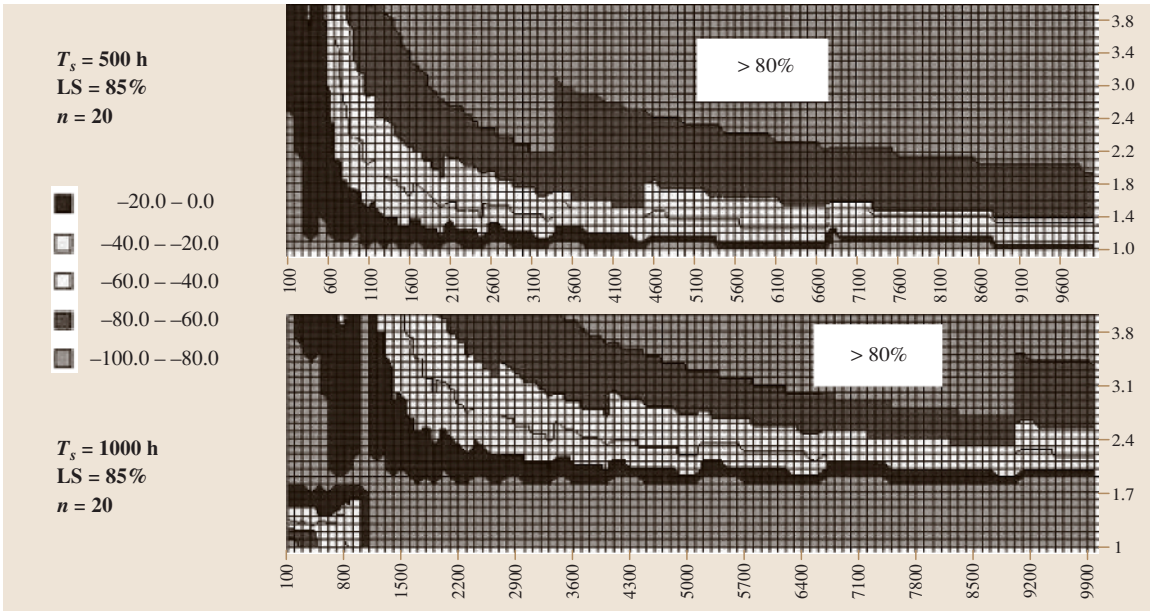


Fig. 48.22 Share of saving with respect to hypothesis of constant failure rate for different parameter values

by (48.29) or (48.44) respectively for given MTBF and β values. For a fixed value of MTBF the saving increases with greater values of β, because the use of the Weibull distribution takes into account that failures are grouped in a specific time region where part breakdowns occur with high probability. The MTBF value of 500 h is very important because it defines T_{residual} equal to zero and so x_2 equals zero for any approach. Values of MTBF lower than 500 h mean that the optimal replacement number is influenced by the quantity x_1, while values grater than 500 h are just defined by the use of quantity x_2 (x_1 being equal to zero). Considering a MTBF range starting from the T_S value for a specific value of β; the saving of spare parts needed decreases with greater values of MTBF, as shown in Fig. 48.23 where two different values of the parameter T_S (500 and 1000 h) are compared. Figure 48.19, as Fig. 48.20, relates the difference between the optimal replacement number calculated by (48.29) and (48.44) for T_S equal to 1500 h.

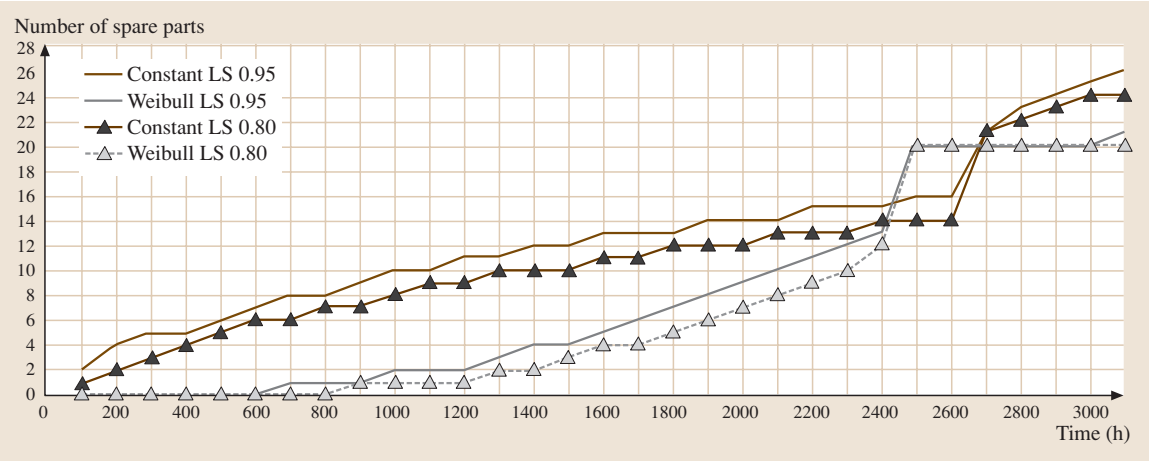


Fig. 48.23 Number of spare parts for component A (number of employments $n = 20$, and $\beta = 4$)

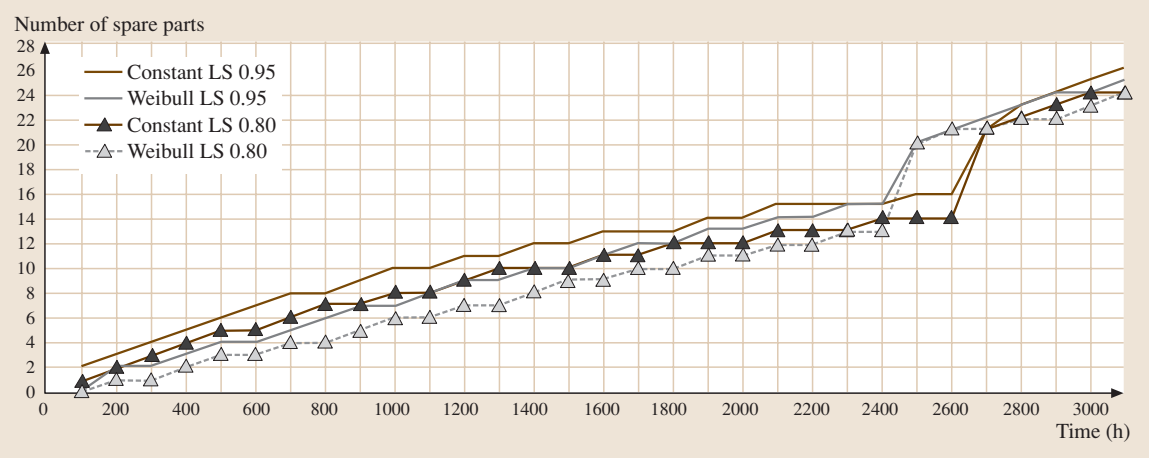


Fig. 48.24 Number of spare parts for component B (number of employments $n = 20$, and $\beta = 1.5$)

Obviously a low frequency of consignments creates a greater requirement for spare parts. Also in this case it is important to notice that methods behave as one when the MTBF is equal to T_S.

48.8.3 Case Study: An Industrial Application

This industrial application deals with an iron metallurgy plant, a European leader in the manufacture of merchant bars. We are interested in production machines characterized by electric iron steel furnaces, rolled sections and the final steps of finishing mills: straightening, cutting to length and packaging.

The extended method was applied with very interesting results, with an average saving of more than 20% in spare parts management. To focus the aim of this study we consider two different components (conic couples), called A and B, that have similar values of MTBF and number of employments n (about 20) but very different values of the parameter β: $\beta = 4$ for part A and $\beta = 1.5$ for part B. The graph in Fig. 48.23 compares the number of spare parts for part A forecasted by methods based on the hypothesis of a time-independent failure rate $\lambda(t)$ or based on the Weibull distribution.

Two different LS values are investigated. It is clear that the use of the Weibull extension reduces the stocks of parts by creating a time delay in the consignments.

The impact will not be significant for values of the β parameter close to one, as stated in Fig. 48.24 for part B, where forecasts are very similar for the methods investigated.

The subject of this study is the evaluation of the spare-parts stock level in maintenance systems in the presence of LTI parts. This paragraph deepens understanding of the fundamental features of a new approach that, in contrast to existing methods, does not consider total spares request to be linear and presents a lower sensitivity to MTBF errors. The proposed model assumes a specific spare-part service level LS defined as the probability of finding the part in case of breakdown: the best values for LS are suggested in Sect. 48.7.1. The aim of the study is to understand whether the hypothesis of a constant failure rate leads to increasing costs or not, compared to more sophisticated distributions. Model extension by the use of the traditional Weibull distribution shows interesting savings in spare parts for values of the β parameter greater than 3 and in the presence of longer times between consignments. The Weibull distribution appears to be a very interesting failure-rate function, but the literature reports some other valid functions; the extended model can easily be extended to any suggested failure-rate function.

References

48.1 A. A. Ghobbar, C. H. Friend: Sources of intermittent demand for aircraft spare parts within airline operations, J. Air Transport Manage. **8**, 221–231 (2002)

48.2 H. S. Lau, M. C. Wang: Estimating the lead-time demand distribution when the daily demand is non-normal and autocorrelated, Eur. J. Oper. Res. **29**, 60–69 (1987)

48.3 J. E. Tyworth, L. O'Neill: Robustness of the normal approximation of lead-time demand in a distribution setting, Naval Res. Logistics **44**, 165–186 (1997)

48.4 J. B. Ward: Determining reorder points when demand is lumpy, Manage. Sci. **24**, 623–632 (1978)

48.5 T. M. Williams: Stock control with sporadic and slow-moving demand, J. Oper. Res. Soc. **35(10)**, 939–948 (1984)

48.6 C. R. Mitchell, R. A. Rappold, W. B. Faulkner: An analysis of Air Force EOQ data with an application to reorder point calculation, Manage. Sci. **29**, 440–446 (1983)

48.7 P. D. Van Ness, W. J. Stevenson: Reorder-point models with discrete probability distributions, Decision Sci. **14**, 363–369 (1983)

48.8 C. R. Schultz: Forecasting and inventory control for sporadic demand under periodic review, J. Oper. Res. Soc. **37**, 303–308 (1987)

48.9 R. B. Watson: The effects of demand-forecast fluctuations on customer service and inventory cost when demand is lumpy, J. Oper. Res. Soc. **38**, 75–82 (1987)

48.10 W. T. M. Dunsmuir, R. D. Snyder: Control of inventories with intermittent demand, Eur. J. Oper. Res. **40**, 16–21 (1989)

48.11 D. Petrovic, R. Petrovic: SPARTA II: Further development in an expert system for advising on stocks of spare parts, Int. J. Prod. Econ. **24**, 291–300 (1992)

48.12 I. J. Bier: *Boeing Commercial Airplane Group Spares Department: Simulation of Spare Parts Operations.* Detroit, MI: ORSA/TIMS Joint National Meeting, (1984)

48.13 B. Sani, B. G. Kingsman: Selecting the best periodic inventory control and demand forecasting methods for low demand items, J. Oper. Res. Soc. **48**, 700–713 (1997)

48.14 T. R. Willemain, C. N. Smart, H. F. Schwarz: A new approach to forecasting intermittent demand for service parts inventories (in press), Int. J. Forecasting **20**(3), 375–387 (2003)

48.15 J. D. Croston: Forecasting and stock control for intermittent demands, Oper. Res. Quart. **23**, 289–303 (1972)

48.16 A. Segerstedt: Inventory control with variation in lead times, especially when demand is intermittent, Int. J. Prod. Econ. **35**, 365–372 (1994)

48.17 F. R. Johnston, J. E. Boylan: Forecasting for items with intermittent demand, J. Oper. Res. Soc. **47**, 113–121 (1996)

48.18 J. H. Bookbinder, A. E. Lordahl: Estimation of inventory re-order levels using the bootstrap statistical procedure, IIE Trans. **21**, 302–312 (1989)

48.19 M. Wang, S. S. Rao: Estimating reorder points and other management science applications by bootstrap procedure, Eur. J. Oper. Res. **56**, 332–342 (1992)

48.20 Y. B. Kim, J. Haddock, T. R. Willemain: The binary bootstrap: Inference with autocorrelated binary data, Commun. Stat. Simul. Comput. **22**, 205–216 (1993)

48.21 D. Park, T. R. Willemain: The threshold bootstrap and threshold jackknife, Comput. Statist. Data Anal. **31**, 187–202 (1999)

48.22 A. A. Ghobbar, C. H. Friend: Evaluation of forecasting methods for intermittent parts demand in the field of aviation: A predictive model, Comp. Oper. Res. **30**(14), 2097–2114 (2003)

48.23 E. Bartezzaghi, R. Verganti, G. Zotteri: A simulation framework for forecasting uncertain lumpy demand, Int. J. Prod. Econ. **59**, 499–510 (1999)

48.24 A. Pareschi, A. Persona, A. Regattieri, E. Ferrari: TPM: A total approach for industrial maintenance question. In: *Proc. 7th Int. Conf. Reliability and Quality in Design ISSAT (2001), Washington D.C.*, ed. by H. Pham, M. Lu, pp. 262–268

48.25 A. Pareschi, A. Persona, A. Regattieri: Methodology for spare parts level optimization in maintenance systems. In: *Proc. ISSAT (2000), Orlando, Florida USA*, ed. by H. Pham, M. Lu

48.26 A. Pareschi, A. Persona, A. Regattieri, E. Ferrari: TPM: Situation and procedure for a soft introduction in italian factories, TQM Mag. **14**(6), 350–358 (2002)

48.27 A. Persona, A. Regattieri, M. Catena: Low turnaround spare parts level optimization. In: *Proc. 9th Int. Conf. Reliability and Quality in Design, ISSAT (2003), Honolulu (Hawaii)*, ed. by H. Pham, M. Lu, pp. 273–277

48.28 A. Syntetos: Forecasting of Intermittent Demand. Ph.D. Thesis (Buckinghamshire Business School, Brunel University, London 2001)

48.29 H. S. Campbell: *The Relationship of Resource Demands to Air Base Operations* (RAND, Santa Monica 1963)

48.30 M. D. Clarke: Irregular airline operations: A review of the state of the practice in airline operations control centres, J. Air Transport Management **4**(2), 67–76 (1998)

48.31 C. H. Friend: *Aircraft Maintenance Management* (Longman, Harlow 1992)

48.32 R. G. Brown: *Statistical Forecasting for Inventory Control* (McGraw–Hill, New York 1959)

48.33 P. Alstrom, P. Madsen: Tracking signals in inventory control systems. A simulation study, Int. J. Prod. Econ. **45**, 293–302 (1996)

48.34 R. Coughlin: Optimization of spares in maintenance scenario. Proc. Reliability and Mainteinability Symposium (1984)

48.35 A. Metaweh: A cost reliability model with spares in electric power system. Proc. Reliability and Maintainability Symp. (1997)

48.36 S. Makridakis, S. C. Wheelwright: *Forecasting Methods for Management* (Wiley, New York 1998)

48.37 U. Hjorth: A reliability distribution with increasing, decreasing, constant and bathtub-shaped failure rates, Technometrics **12**, 22–99 (1980)

48.38 G. S. Mudholkar, D. K. Srivastava: Exponentiated Weibull family for analyzing bathtube failure-rate data, IEEE Trans. Reliab. **44**, 388–91 (1993)

48.39 Z. Chen: A new two-parameter lifetime distribution with bathtube shape or increasing failure rate function, Stat. Prob. Lett. **49**, 155–161 (2000)

48.40 M. Xie, Y. Tang, T. N. Goh: A modified Weibull extension with bathtub-shaped failure rate function, Reliab. Eng. Syst. Safety **76**(3), 279–285 (2002)

49. Arithmetic and Geometric Processes

Section 49.1 introduces two special monotone processes. A stochastic process is an AP (or a GP) if there exists some real number (or some positive real number) such that after some additions (or multiplications) it becomes a renewal process (RP). Either is a stochastically monotonic process and can be used to model a point process, i. e. point events occurring in a haphazard way in time or space, especially with a trend. For example, the events may be failures arising from a deteriorating machine, and such a series of failures is distributed haphazardly along a time continuum.

Sections. 49.2–49.5 discuss estimation procedures for a number K of independent, homogeneous APs (or GPs). More specifically; in Sect. 49.2, Laplace's statistics are recommended for testing whether a process has a trend or K processes have a common trend, and a graphical technique is suggested for testing whether K processes come from a common AP (or GP) as well as having a common trend; in Sect. 49.3, three parameters – the common difference (or ratio), the intercept and the variance of errors – are estimated using simple linear regression techniques; in Sect. 49.4, a statistic is introduced for testing whether K processes come from a common AP (or GP); in Sect. 49.5, the mean and variance of the first average random variable of the AP (or GP) are estimated based on the results derived in Sect. 49.3.

Section 49.6 mentions some simulation studies performed to evaluate various nonparametric estimators and to compare the estimates, obtained from various estimators, of the parameters. Some suggestions for selecting the best estimators under three non-overlapping ranges of the common difference (or ratio) values are made based on the results of the simulation studies.

In Sect. 49.7, ten real data sets are treated as examples to illustrate the fitting of AP, GP, homogeneous Poisson process (HPP) and nonhomogeneous Poisson process (NHPP) models.

In Sect. 49.8, new repair–replacement models are proposed for a deteriorating system, in which the successive operating times of the system form an arithmetico-geometric process (AGP) and are stochastically decreasing, while the successive repair times after failure also constitute an AGP but are stochastically increasing. Two kinds of replacement policy are considered, one based on the working age (a continuous decision variable) of the system and the other determined by the number of failures (a discrete decision variable) of the system. These policies are considered together with the performance measures, namely loss (or its negation, profit), cost, and downtime (or its complement, availability). Applying the well-known results of renewal reward processes, expressions are derived for the long-run expected performance measure per unit total time, and for the long-run expected performance measure per unit operation time, under the two kinds of policy proposed.

In Sect. 49.9, some conclusions of the applicability of an AP and/or a GP based on partial findings of four real case studies are drawn.

Section 49.10 gives five concluding remarks. Finally, the derivations of some key results are outlined in the Appendix, followed by the results of both the APs and GPs summarized in Table 49.6 for easy reference.

Most of the content of this chapter is based on the author's own original works that appeared in *Leung* et al. [49.1–13], while some is extracted from *Lam* et al. [49.14–16].

In this chapter, the procedures are, for the most part, discussed in reliability terminology. Of course, the methods are valid in any area of application (see Examples 1, 5, 6 and 9 in Sect. 49.7), in which case they should be interpreted accordingly.

In the statistical analysis of a series of events, a common method is to model the series using a point process. To start with, it is essential to test whether the data of successive inter-event times, denoted by X_i $(i = 1, 2, \ldots)$, demonstrate a trend. If there is no trend, we may model the data using a stationary point process (i. e. a counting process that has stationary, but not necessarily independent, increments), or using a sequence of independent and identically distributed (i.i.d.) random variables $X \equiv X_i$ for all i. For the latter, we may model the corresponding counts of events in time using a renewal process (RP). In particular, if X is exponentially distributed with a rate parameter λ, we may use a homogeneous Poisson process (HPP) with a constant rate λ to model the data. The HPP is one of the most common stochastic processes for modeling counts of events in time or area/volume. This process is a standard for *randomness*, as the assumptions involved state that events must occur independently and any two non-overlapping intervals of the same size have the same probability of capturing one of the events of interest. However, in practice the data of successive inter-event times usually exhibit a trend. We may model them using a nonstationary model, or using a nonhomogeneous Poisson process (NHPP) in which the rate at time t is a function of t. The NHPP is a popular approach used to model data with a trend. For more details of these methods, see *Cox* and *Lewis* [49.17], and *Ascher* and *Feingold* [49.18].

Most research on the maintenance of a repairable system has made either the perfect or minimal repair assumption. Perfect repair means that, after repair, a failed system is as good as new, i. e. a system's successive operating times constitute an RP, see *Barlow* and *Proschan* [49.19]. For a perfect repair model, if the time needed to repair a system is considered negligible, results of RPs can be applied to resolve the system's maintenance problems, see *Ross* [49.20]; if repair time has to be taken into account and the corresponding consecutive repair times constitute another RP, results of alternating RPs can be applied instead, see *Birolini* [49.21]. However, in practice, this is not always the case. Minimal repair means that a failed system will function, after repair, with the same rate of failure and the same effective age as at the epoch of the last failure. For a minimal repair model, where repair time is assumed negligible, an NHPP in which the rate of occurrence of failures (ROCOF) is monotone can provide at least a good first-order model for a deteriorating system, see *Ascher* and *Feingold* [49.18]. That is, failures constitute an NHPP with a suitable parametric form for ROCOF. If the repair time has to be taken into account, the NHPP approach cannot be used.

The popularity of the power-law process (PLP) is based on two features: firstly, it can model deteriorating or improving systems; secondly, point estimators for the parameters have simple closed-form expressions and hypotheses tests can be undertaken using existing tables. The PLP denoted and given by $r(t) = \lambda\beta t^{\beta-1}$ for $t \geq 0$ and λ, $\beta > 0$ is the most important ROCOF parametric form in an NHPP model. If $\beta > 1$, the ROCOF increases

with time, as often happens with aging machinery. This is one of the two main conditions for a preventive replacement being worth carried out (the other condition is that the average cost c_p of a replacement is much greater than that c_r of a minimal repair). But if $0 < \beta < 1$, the ROCOF decreases with time; hence, the PLP can model reliability growth as well. The HPP, which has constant ROCOF, is a special case of the PLP with $\beta = 1$. *Rigdon* and *Basu* [49.22] gave a detailed discussion of the PLP.

Another two-parameter ROCOF form, widely quoted in the literature, that can also model deterioration and reliability growth is the log-linear process (LLP), in which the ROCOF at time t is modeled as $s(t) = e^{\alpha_0+\alpha_1 t}$ for $t \geq 0$ and $-\infty < \alpha_0, \alpha_1 < \infty$, with $\alpha_1 > 0$ under deterioration. This process is less often used, possibly because it is seldom found to be applicable or mathematically less tractable.

Barlow and *Hunter* [49.23] first introduced the idea of minimal repair and proposed a system which is replaced with a regular period T and undergoes minimal repairs upon failures between the periodic replacements. *Muth* [49.24] then proposed a replacement model in which minimal repairs upon system failures are performed up to age T and the system is replaced at the first failure after T. Later, *Park* [49.25] proposed a modification of the model in which a system undergoes minimal repairs for the first $(N-1)$ failures and is replaced at the Nth failure. *Nakagawa* and *Kowada* [49.26] put the first and third types of policy together and constructed a replacement model in which a system is regularly replaced with a period T or at the Nth failure after its installation, whichever occurs first. The system undergoes only minimal repair upon failures between the periodic replacements. In *Leung* and *Cheng* [49.3], Nakagawa and Kowada's replacement model was employed, and the optimal replacement policy based on minimizing the long-run expected cost per month for each type of engine was determined.

The perfect repair model may be reasonable for a system with one simple unit only, and the minimal repair model seems plausible for systems consisting of many components, each having its own failure mode. In many practical instances, repair activities may not result in such extreme situations but in complicated intermediate ones. *Brown* and *Proschan* [49.27] considered the model of imperfect repair which, with probability p, is a perfect repair or, with probability $1-p$, is a minimal repair. *Kijima* [49.28, 29] studied a more general repair model which includes the imperfect repair model as a special case. For a review of imperfect maintenance models, see *Pham* and *Wang* [49.30], and *Wang* and *Pham* [49.31].

An arithmetic process (AP) or a geometric process (GP), which is a nonstationary model, can be used as an alternative to the NHPP in analyzing data of inter-event times that exhibit a trend. This appears to be a useful model for failure or repair data arising from a single system or a collection of independent, homogeneous systems. Consider the maintenance problems of a repairable system and bear in mind that most repairable systems, like engines and gearboxes, are deteriorative. Two basic characteristics of a deteriorating system are that, because of wear through operation or metal fatigue under stress, the system's successive operating times decrease and so the system's life is finite; and that, because it is more difficult and hence takes more time to rectify accumulated wear, the corresponding consecutive repair times increase until finally the system is beyond repair. Based on this understanding, an AP approach proposed by *Leung* [49.5] or a GP approach proposed by *Lam* [49.14] is considered more relevant, realistic and direct for the modeling of maintenance problems in a deteriorating system. Although all discussions in this chapter are in terms of deteriorating systems, they are also valid for improving systems (see Examples 2, 4 and 10 in Sect. 49.7).

The following main symbols in the text are adopted. For a fixed $k = 1, \ldots, K$,

- $A_{n,k}$ (or $G_{n,k}$) denotes either the operating time after the $(n-1)$th repair for $n = 1, 2, \ldots, N_k$ with $X_0 = 0$, or the repair time after the nth failure for $n = 1, 2, \ldots, N_k$;
- d denotes either a common difference d_a of a decreasing arithmetic process such that $d_a \in \left(0, \frac{\mu_{A_1,k}}{n-1}\right]$ or a common difference d_b of an increasing arithmetic process such that $d_b < 0$;
- r denotes either a common ratio r_a of a decreasing geometric process such that $r_a > 1$ or a common ratio r_b of an increasing geometric process such that $0 < r_b < 1$;
- $\mu_{A_{n,k}}$ (or $\mu_{G_{n,k}}$) is the mean of $A_{n,k}$ (or $G_{n,k}$) for $n = 1, 2, \ldots, N_k$;
- $\sigma^2_{A_{n,k}}$ (or $\sigma^2_{G_{n,k}}$) is the variance of $A_{n,k}$ (or $G_{n,k}$) for $n = 1, 2, \ldots, N_k$;
- $\varepsilon_{A,n,k}$ (or $\varepsilon_{G,n,k}$) is an error term with mean 0 and constant variance denoted by $\sigma^2_{A,\varepsilon}$ (or $\sigma^2_{G,\varepsilon}$).

49.1 Two Special Monotone Processes

The work in this section is substantially based on *Leung* [49.4, 11].

49.1.1 Arithmetic Processes

Suppose that K independent, homogeneous APs are available. A definition of the kth AP for $k = 1, \ldots, K$ is given below.

Definition 49.1
Given a sequence of random variables $A_{1,k}, A_{2,k}, \ldots$, if for some real number d, $\{A_{n,k} + (n-1)d, n = 1, 2, \ldots\}$ forms an RP, then $\{A_{n,k}, n = 1, 2, \ldots\}$ is an AP for $k = 1, \ldots, K$. The constant d is called the common difference of the AP.

Three specializations of an AP are given below.

If $d \in \left(0, \frac{\mu_{A_1}}{n-1}\right]$, where $n = 2, 3, \ldots$ and $k = 1, \ldots, K$; and $\mu_{A_1,k}$ is the mean of the first random variable $A_{1,k}$, then the AP is called a decreasing AP. If $d < 0$, then the AP is called an increasing AP. If $d = 0$, then the AP reduces to an RP.

The upper bound of d in the first specialization can be obtained as follows: by Definition 49.1, the expression for the general term of an AP is given by $A_{n,k} \stackrel{d}{=} A_{1,k} - (n-1)d$. Taking expectations of both sides of this expression, and remembering that $A_{n,k}$ is a nonnegative random variable and hence $\boldsymbol{E}(A_{n,k}) \equiv \mu_{A_n,k} \geq 0$ for $n = 1, 2, \ldots$; we obtain, after transposition, the upper bound of d given by $\frac{\mu_{A_1,k}}{n-1}$ for $n = 2, 3, \ldots$. Clearly, the positive integer n is limited for a decreasing AP. Moreover, if the value of d is close to its upper bound, we will obtain a short sequence of nonnegative random variables. However, such a subtractive process is likely to be useful in a deteriorating system (e.g. an engine or a gearbox), which fails rarely (e.g. two/three times) over its usual span of life (e.g. five years). This implicitly means that the system wears out, between two successive failures, to such an extraordinary extent that the corresponding system's successive operating time decreases dramatically.

Given an AP $\{A_{n,k}, n = 1, 2, \ldots\}$ for $k = 1, \ldots, K$, we have $A_{n,k} \stackrel{d}{=} A_{1,k} - (n-1)d$ by Definition 49.1. Therefore, the means and variances of $A_{n,k}$ can respectively be written as

$$\begin{aligned}\mu_{A_n,k} &\equiv \boldsymbol{E}(A_{n,k}) = \boldsymbol{E}(A_{1,k}) - (n-1)d \\ &\equiv \mu_{A_1,k} - (n-1)d \end{aligned} \tag{49.1}$$

and

$$\begin{aligned}\sigma^2_{A_n,k} &\equiv \boldsymbol{V}(A_{n,k}) = \boldsymbol{V}[A_{1,k} - (n-1)d] = V(A_{1,k}) \\ &\equiv \sigma^2_{A_1,k} \,. \end{aligned} \tag{49.2}$$

Consider K independent, homogeneous APs $\{A_{n,k}, n = 1, \ldots, N_k$ and $k = 1, \ldots, K\}$ together. Without loss of generality, we assume $N_1 \geq N_2 \geq \ldots \geq N_K$. Denote

$$\bar{A}_n \equiv \frac{\sum_{k=1}^{K^*} A_{n,k}}{K^*} \,,$$

$$\mu_{\bar{A}_1} \equiv \frac{\sum_{k=1}^{K} \mu_{A_1,k}}{K} = \frac{\sum_{k=1}^{K^*} \mu_{A_1,k}}{K^*} \,, \quad \text{and}$$

$$\sigma^2_{\bar{A}_1} \equiv \frac{\sum_{k=1}^{K} \sigma^2_{A_1,k}}{K} = \frac{\sum_{k=1}^{K^*} \sigma^2_{A_1,k}}{K^*} \,,$$

where

$$\begin{aligned} K^* &= K && \text{for } n = 1, \ldots, N_K \text{ or,} \\ &= K-1 && \text{for } n = N_K + 1, \ldots, N_{K-1} \text{ or,} \\ &= K-2 && \text{for } n = N_{K-1} + 1, \ldots, N_{K-2} \text{ or,} \\ &\vdots \\ &= 1 && \text{for } n = N_2 + 1, \ldots, N_1 \,. \end{aligned}$$

To clarify the definition of $\bar{A}_n$, let us consider $K = 3$, $N_1 = 5$, $N_2 = 3$, $N_3 = 2$. Then we have

$$\begin{aligned} \bar{A}_1 &= \frac{A_{1,1} + A_{1,2} + A_{1,3}}{3} \,, \\ \bar{A}_2 &= \frac{A_{2,1} + A_{2,2} + A_{2,3}}{3} \,, \\ \bar{A}_3 &= \frac{A_{3,1} + A_{3,2}}{2} \,, \\ \bar{A}_4 &= A_{4,1} \text{ and } \bar{A}_5 = A_{5,1} \,. \end{aligned}$$

Therefore, using (49.1) and (49.2), the means and variances of $\bar{A}_n$ can respectively be written as

$$\begin{aligned}\mu_{\bar{A}_n} &\equiv \boldsymbol{E}(\bar{A}_n) = \frac{\sum_{k=1}^{K^*} E(A_{n,k})}{K^*} \\ &= \frac{\sum_{k=1}^{K^*} E[A_{1,k} - (n-1)d)]}{K^*} \\ &\equiv \frac{\sum_{k=1}^{K^*} \mu_{A_1,k} - K^*(n-1)d}{K^*} = \mu_{\bar{A}_1} - (n-1)d\end{aligned} \tag{49.3}$$

and

$$\begin{aligned}\sigma^2_{\bar{A}_n} &\equiv \boldsymbol{V}(\bar{A}_n) = \frac{\sum_{k=1}^{K^*} V(A_{n,k})}{(K^*)^2} \\ &= \frac{\sum_{k=1}^{K^*} V[A_{1,k} - (n-1)d]}{(K^*)^2} \\ &\equiv \frac{\sum_{k=1}^{K^*} \sigma^2_{A_1,k}}{(K^*)^2} = \frac{\sigma^2_{\bar{A}_1}}{K^*} .\end{aligned} \tag{49.4}$$

49.1.2 Geometric Processes

Suppose that K independent, homogeneous GPs are available. A definition of the kth GP for $k = 1, \ldots, K$ is given below.

Definition 49.2

Given a sequence of random variables $G_{1,k}, G_{2,k}, \ldots$, if for some $r > 0$, $\{r^{(n-1)}G_{n,k}, n = 1, 2, \ldots\}$ forms an RP, then $\{G_{n,k}, n = 1, 2, \ldots\}$ is a GP for $k = 1, \ldots, K$. The constant r is called the common ratio of the GP.

Three specializations of a GP are given below.

If $r > 1$, then the GP is called a decreasing GP. If $0 < r < 1$, then the GP is called an increasing GP. If $r = 1$, then the GP reduces to an RP.

Given a GP $\{G_{n,k}, n = 1, 2, \ldots\}$ for $k = 1, \ldots, K$; we have $G_{n,k} \stackrel{d}{=} \frac{G_{1,k}}{r^{(n-1)}}$ from Definition 49.2. Therefore, the means and variances of $G_{n,k}$ can be written as

$$\mu_{G_n,k} \equiv \boldsymbol{E}(G_{n,k}) = \frac{\boldsymbol{E}(G_{1,k})}{r^{(n-1)}} \equiv \frac{\mu_{G_1,k}}{r^{(n-1)}} \tag{49.5}$$

and

$$\begin{aligned}\sigma^2_{G_n,k} &\equiv \boldsymbol{V}(G_{n,k}) = \boldsymbol{V}\left(\frac{G_{1,k}}{r^{(n-1)}}\right) = \frac{\boldsymbol{V}(G_{1,k})}{r^{2(n-1)}} \\ &\equiv \frac{\sigma^2_{G_1,k}}{r^{2(n-1)}} .\end{aligned} \tag{49.6}$$

Consider K independent, homogeneous GPs $\{G_{n,k}, n = 1, \ldots, N_k \text{ and } k = 1, \ldots, K\}$ together. Without loss of generality, we assume $N_1 \geq N_2 \geq \ldots \geq N_K$. Denote

$$\begin{aligned}\bar{G}_n &\equiv \frac{\sum_{k=1}^{K^*} G_{n,k}}{K^*} , \\ \mu_{\bar{G}_1} &\equiv \frac{\sum_{k=1}^{K} \mu_{G_1,k}}{K} = \frac{\sum_{k=1}^{K^*} \mu_{G_1,k}}{K^*} , \quad \text{and} \\ \sigma^2_{\bar{G}_1} &\equiv \frac{\sum_{k=1}^{K} \sigma^2_{G_1,k}}{K} = \frac{\sum_{k=1}^{K^*} \sigma^2_{G_1,k}}{K^*} ,\end{aligned}$$

where K^* has previously been defined and $\bar{G}_n$ has a similar meaning to $\bar{A}_n$.

Therefore, using (49.5) and (49.6), the means and variances of $\bar{G}_n$ can respectively be written as

$$\begin{aligned}\mu_{\bar{G}_n} &\equiv \boldsymbol{E}(\bar{G}_n) = \frac{\sum_{k=1}^{K^*} \boldsymbol{E}(G_{n,k})}{K^*} = \frac{\sum_{k=1}^{K^*} \boldsymbol{E}\left[\frac{G_{1,k}}{r^{(n-1)}}\right]}{K^*} \\ &\equiv \frac{\sum_{k=1}^{K^*} \frac{\mu_{G_1,k}}{r^{(n-1)}}}{K^*} = \frac{\mu_{\bar{G}_1}}{r^{(n-1)}}\end{aligned} \tag{49.7}$$

and

$$\sigma^2_{\bar{G}_n} \equiv \boldsymbol{V}(\bar{G}_n) = \boldsymbol{V}\left(\frac{\sum_{k=1}^{K^*} G_{n,k}}{K^*}\right) = \frac{\sum_{k=1}^{K^*} V(G_{n,k})}{(K^*)^2} ,$$

as $G_{n,k}$s are independent

$$= \frac{\sum_{k=1}^{K^*} V\left[\frac{G_{1,k}}{r^{(n-1)}}\right]}{(K^*)^2} \equiv \frac{\sum_{k=1}^{K^*} \frac{\sigma^2_{G_1,k}}{r^{2(n-1)}}}{(K^*)^2} = \frac{\sigma^2_{\bar{G}_1}}{r^{2(n-1)}K^*} . \tag{49.8}$$

According to the three specializations of an AP (or a GP), for a deteriorating system, it is reasonable to assume that the successive operating times of the system form a decreasing AP (or GP), whereas the corresponding consecutive repair times constitute an increasing AP

(or GP). However, the replacement times for the system are usually stochastically the same no matter how old the used system is; hence, these will form an RP. This is the motivation behind the introduction of the AP (or GP) approach.

Thus, d, $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ (or r, $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$) are the most important parameters in K APs (or GPs) because the means and variances of the $\bar{A}_n$s (or $\bar{G}_n$s) are completely determined by these three parameters. In view of this fact, in this chapter the procedure is defined for applying the AP (or GP) approach in a reliability context and the functions of estimators are derived for the three fundamental parameters. Now, there are three questions. The first is, given a set of data of successive inter-event times of a point process, how do we test whether this is consistent with an AP (or a GP)? The second question is, if the data do come from a common AP (or GP), how can we estimate the parameters d, $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ (or r, $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$)? The third question is, after fitting an AP (or a GP) model to the data set, how good is the fit?

In this chapter, the statistical inference for K independent, homogeneous APs (or GPs) is investigated and the first two questions are answered using well-known statistical methods. In Sect. 49.3, the parameters d, α_{A} and $\sigma^2_{\mathrm{A},\varepsilon}$ (or r, α_{G} and $\sigma^2_{\mathrm{G},\varepsilon}$) are estimated using simple linear regression techniques. In Sect. 49.5, first $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ (or $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$) are estimated based on the results derived in Sect. 49.3, and then $\mu_{\bar{A}_n}$ and $\sigma^2_{\bar{A}_n}$ (or $\mu_{\bar{G}_n}$ and $\sigma^2_{\bar{G}_n}$) are correspondingly estimated using (49.3) and (49.4) [or (49.7) and (49.8)], respectively.

49.2 Testing for Trends

Much of the work in this section is based on *Leung* [49.4, 11].

49.2.1 Laplace Test

Suppose now that K independent, homogeneous series are available with periods of observation $T_1, T_2, \ldots, T_K$. The numbers of events in the different series are denoted by $N_1, N_2, \ldots, N_K$ and the times of occurrence of events by $Y_{n,1}, Y_{n,2}, \ldots, Y_{n,K}$. Given the data $\{A_{n,k}$ (or $G_{n,k}), n = 1, 2, \ldots$ and $k = 1 \ldots, K\}$ of successive inter-event times of a point process, first of all we need to test whether the $A_{n,k}$s are identically distributed by checking for the existence of a trend. To do this, many techniques discussed in *Ascher* and *Feingold* [49.18] can be used. Laplace's trend test is used for ease of manipulation and interpretation.

Null hypothesis H_0: $A_{n,k}$s (or $G_{n,k}$) are identically distributed.

Alternative hypothesis H_1: $A_{n,k}$s (or $G_{n,k}$) are not identically distributed, i. e. there is a trend.

Laplace's test statistic for a time-truncated data set (i. e. when the data are time truncated, the time of the conclusion of observation is fixed and the number of events is random) is given by

$$L_k = \frac{\sum_{n=1}^{N_k} Y_{n,k} - \frac{N_k T_k}{2}}{\sqrt{\frac{N_k T_k^2}{12}}} \quad \text{for } k = 1, \ldots, K\,, \tag{49.9}$$

where $Y_{1,k}, \ldots, Y_{N_k,k}$, with $Y_{n,k} = \sum_{i=1}^{n} A_{i,k}$ (or $Y_{n,k} = \sum_{i=1}^{n} G_{i,k}$) are the event times for a process observed in $(0, T_k]$, and T_k is the pre-specified time of observation.

Laplace's test statistic for an event-truncated data set (i. e. when the data are event-truncated, the number of events is fixed before observation begins and the time of the conclusion of the observation is random) is given by

$$L_k = \frac{\sum_{n=1}^{N_k-1} Y_{n,k} - \frac{(N_k-1)Y_{N_k,k}}{2}}{\sqrt{\frac{(N_k-1)Y^2_{N_k,k}}{12}}} \quad \text{for } k = 1, \ldots, K\,, \tag{49.10}$$

where $Y_{n,k} = \sum_{i=1}^{n} A_{i,k}$ (or $Y_{n,k} = \sum_{i=1}^{n} G_{i,k}$) is the time of the nth failure for $n = 1, 2, \ldots, N_k$, and N_k is the pre-specified number of events.

L_k is approximately distributed as the standard normal for $N_k \geq 3$, time-truncated data, or $N_k \geq 4$, event-truncated data, at the 5% level of significance, see *Ascher* and *Feingold* [49.18]. If $|L_k| > 1.96$, then H_0 is rejected at the 5% level of significance, i. e. the event data set $\{A_1, A_2, \ldots, A_{N_k}\}$ (or $\{G_1, G_2, \ldots, G_{N_k}\}$) exhibits a trend.

Rigdon and *Basu* [49.22], on p. 259, reach the conclusion that

> *using any model for event times, one clearly indicates the time that data collection started and the time that it ceased. This is necessary so that the appropriate analysis, that is, an analysis based on event-truncated or time-truncated data, can be applied and maximum information can be obtained from the data. For time-truncated data, the time between the last event and the termination of the test contains some information that should not be wasted.*

Suppose, however, that a pooled test is required. It would often be best to take the null hypothesis to be that the series individually follow stationary point processes which possibly differ for different series. We can then make a combined test for trend in the data and this can be done using (49.11) or (49.12), which are given below.

Laplace's test pooled statistic for a time-truncated data set is given by

$$L = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} Y_{n,k} - \frac{1}{2}\sum_{k=1}^{K} N_k T_k}{\sqrt{\frac{\sum_{k=1}^{K} N_k T_k^2}{12}}} \qquad (49.11)$$

and Laplace's test pooled statistic for an event-truncated data set is given by

$$L = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k-1} Y_{n,k} - \frac{1}{2}\sum_{k=1}^{K}(N_k-1)Y_{N_k,k}}{\sqrt{\frac{\sum_{k=1}^{K}(N_k-1)Y_{N_k,k}^2}{12}}}\,, \qquad (49.12)$$

which under the null hypothesis has zero mean, unity variance and very nearly a normal distribution.

Note that the series can be tested individually for trend using (49.9) or (49.10); however, it is worth making a combined trend test in the data using (49.11) or (49.12). *Cox* and *Lewis* [49.17], on p. 50, note that there are other ways the separate trend tests could be combined, for example by forming

$$L = \frac{\sum_{k=1}^{K} L_k}{\sqrt{12}} \quad \text{or} \quad L = \sum_{k=1}^{K} L_k^2$$

the former would be tested as a standardized normal variable, the latter as chi-squared with $(K-1)$ degrees of freedom. These tests take no account of the very different numbers of observations in the different series.

49.2.2 Graphical Techniques

Another possible approach is to use simple linear regression techniques.

Arithmetic Processes

To start with, let

$$W_{A,n,k} = A_{n,k} + (n-1)d\,. \qquad (49.13)$$

From Definition 49.1, $W_{A,n,k}$s are i.i.d. and can be written as

$$W_{A,n,k} = \alpha_A + \varepsilon_{A,n,k}\,, \qquad (49.14)$$

where

$$E(W_{A,n,k}) = \alpha_A \qquad (49.15)$$

and $\varepsilon_{A,n,k}$s are also i.i.d. (not necessarily normally distributed if our objective is estimation only, e.g. see *Gujarati* [49.32], p. 281) with

$$E(\varepsilon_{A,n,k}) = E(\varepsilon_{A,n}) = 0, \text{ irrespective of } k \qquad (49.16)$$

and

$$V(\varepsilon_{A,n,k}) = V(\varepsilon_{A,n}) \equiv \sigma^2_{A,\varepsilon_n}\,, \qquad (49.17)$$

irrespective of k and $\sigma^2_{A,\varepsilon_n} = \sigma^2_{A,\varepsilon}$, equal variance irrespective of n.

Combining (49.13) and (49.14) yields

$$A_{n,k} = -d(n-1) + \alpha_A + \varepsilon_{A,n,k} \quad \text{for } n = 1, \ldots, N_k \quad \text{and } k = 1, \ldots, K\,, \qquad (49.18)$$

which is a simple linear regression equation.

Geometric processes

To start with, let

$$W_{\mathrm{G},n,k} = r^{(n-1)} G_{n,k} \tag{49.19}$$

or

$$\ln W_{\mathrm{G},n,k} = (n-1)\ln r + \ln G_{n,k}\,. \tag{49.20}$$

From Definition 49.2, $W_{G,n,k}$s are i.i.d. and can be written as

$$\ln W_{\mathrm{G},n,k} = \alpha_{\mathrm{G}} + \varepsilon_{\mathrm{G},n,k} \tag{49.21}$$

or

$$W_{\mathrm{G},n,k} = \mathrm{e}^{\alpha_{\mathrm{G}}+\varepsilon_{\mathrm{G},n,k}}\,, \tag{49.22}$$

where

$$\mathrm{E}(\ln W_{\mathrm{G},n,k}) = \alpha_{\mathrm{G}} \tag{49.23}$$

and $\varepsilon_{G,n,k}$s are also i.i.d. (not necessarily normally distributed if our objective is estimation only, e.g. see *Gujarati* [49.32], p. 281) with

$$\mathrm{E}(\varepsilon_{\mathrm{G},n,k}) = \mathrm{E}(\varepsilon_{\mathrm{G},n}) = 0,\ \text{irrespective of } k \tag{49.24}$$

and

$$\mathrm{V}(\varepsilon_{\mathrm{G},n,k}) = \mathrm{V}(\varepsilon_{\mathrm{G},n}) \equiv \sigma^2_{\mathrm{G},\varepsilon_n}\,, \tag{49.25}$$

irrespective of k and $\sigma^2_{\mathrm{G},\varepsilon_n} = \sigma^2_{\mathrm{G},\varepsilon}$, equal variance irrespective of n.

Combining (49.20) and (49.21) yields

$$\ln G_{n,k} = -\ln r(n-1) + \alpha_{\mathrm{G}} + \varepsilon_{\mathrm{G},n,k} \quad \text{for } n = 1,\ldots,N_k \quad \text{and } k = 1,\ldots,K\,, \tag{49.26}$$

which is a simple linear regression equation.

According to (49.18) [or (49.26)], we can plot $A_{n,k}$ (or $\ln G_{n,k}$) against $(n-1)$ for $n = 1,\ldots,N_k$ and $k = 1,\ldots,K$ to see whether there is a linear relationship between them. Clearly, this is also useful for testing whether the observations $\{A_{n,k}$ (or $G_{n,k})$, $n = 1,\ldots,N_k$ and $k = 1,\ldots,K\}$ come from a common AP (or GP) as well as whether they share a common trend.

49.3 Estimating the Parameters

The work in this section is substantially based on *Leung* [49.4, 11].

49.3.1 Estimate Parameters d, α_{A} and $\sigma^2_{\mathrm{A},\varepsilon}$ of K APs (or r, α_{G} and $\sigma^2_{\mathrm{G},\varepsilon}$ of K GPs)

We can estimate the parameters d, α_{A} and $\sigma^2_{\mathrm{A},\varepsilon}$ using the simple linear regression method. The least-squares point estimates $\hat{d}$, $\hat{\alpha}_{\mathrm{A}}$ and $\hat{\sigma}^2_{\mathrm{A},\varepsilon}$ of the parameters d, α_{A} and $\sigma^2_{\mathrm{A},\varepsilon}$ are calculated respectively using the following formulae:

$$\hat{d} = \frac{\frac{\left(\sum_{k=1}^{K} N_k^2 - N\right)\left(\sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k}\right)}{2N} - \sum_{k=1}^{K}\sum_{n=1}^{N_k}(n-1)A_{n,k}}{\sum_{k=1}^{K}\frac{(N_k-1)N_k(2N_k-1)}{6} - \frac{\left(\sum_{k=1}^{K} N_k^2 - N\right)^2}{4N}}\,, \tag{49.27}$$

$$\hat{\alpha}_{\mathrm{A}} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k}}{N} + \frac{\hat{d}\left(\sum_{k=1}^{K} N_k^2 - N\right)}{2N} \tag{49.28}$$

and

$$\hat{\sigma}^2_{\mathrm{A},\varepsilon} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k}^2 - \frac{\left(\sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k}\right)^2}{N}}{N-2} + \frac{\hat{d}\left(\sum_{k=1}^{K}\sum_{n=1}^{N_k}(n-1)A_{n,k}\right)}{N-2} - \frac{\hat{d}\left(\frac{\left(\sum_{k=1}^{K} N_k^2 - N\right)\left(\sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k}\right)}{2N}\right)}{N-2}\,, \tag{49.29}$$

where

$$N = \sum_{k=1}^{K} N_k\,. \tag{49.30}$$

The derivations of (49.27) to (49.29) are given in the Appendix.

The least-squares point estimates $\hat{r}$, $\hat{\alpha}_{\mathrm{G}}$ and $\hat{\sigma}^2_{\mathrm{G},\varepsilon}$ of the parameters r, α_{G} and $\sigma^2_{\mathrm{G},\varepsilon}$ can be obtained simply

by replacing $\hat{d}$ with $\ln \hat{r}$ and $A_{n,k}$ with $\ln G_{n,k}$ on the right-hand side of (49.27) to (49.29).

49.3.2 Estimating the Parameters of a Single AP (or GP)

When $K = 1$, (49.30) becomes $N = N_1$. Note that

1. for a single AP or GP, we simply use N to represent the number of successive events, and
2. the equations given below are consistent with those derived in *Leung* [49.8] and *Lam* [49.15].

Then (49.27) to (49.29) become

$$\hat{d} = \frac{6(N-1)\sum_{n=1}^{N} A_n - 12\sum_{n=1}^{N}(n-1)A_n}{(N-1)N(N+1)}\,, \tag{49.27.1}$$

$$\hat{\alpha}_{\mathrm{A}} = \frac{2(2N-1)\sum_{n=1}^{N} A_n - 6\sum_{n=1}^{N}(n-1)A_n}{N(N+1)} \tag{49.28.1}$$

and

$$\hat{\sigma}^2_{\mathrm{A},\varepsilon} = \frac{\sum_{n=1}^{N} A_n^2 - \frac{1}{N}\left(\sum_{n=1}^{N} A_n\right)^2}{N-2} - \frac{\hat{d}\left[\frac{(N-1)}{2}\sum_{n=1}^{N} A_n - \sum_{n=1}^{N}(n-1)A_n\right]}{N-2}\,. \tag{49.29.1}$$

For a single GP, $\hat{d}$, $\hat{\alpha}_{\mathrm{A}}$, $\hat{\sigma}^2_{\mathrm{A},\varepsilon}$ and A_n are replaced by $\ln \hat{r}$, $\hat{\alpha}_{\mathrm{G}}$, $\hat{\sigma}^2_{\mathrm{G},\varepsilon}$ and $\ln G_n$ in (49.27.1) to (49.29.1).

49.4 Distinguishing a Renewal Process from an AP (or a GP)

Much of the work in this section is based on *Leung* [49.4, 11]. We test whether the data comes from an RP or AP.

Null hypothesis $\mathbf{H_0}$: $d = 0$

Alternative hypothesis $\mathbf{H_1}$: $d \neq 0$

The t-test statistic is denoted and given by

$$t_{\mathrm{A}} = \frac{-\hat{d}\sqrt{\sum_{k=1}^{K}\frac{(N_k-1)N_k(2N_k-1)}{6} - \frac{\left(\sum_{k=1}^{K} N_k^2 - N\right)^2}{4N}}}{\hat{\sigma}_{\mathrm{A},\varepsilon}}\,, \tag{49.31}$$

where t_{A} is distributed as a Student's t with $(N-2)$ degrees of freedom. If $|t_{\mathrm{A}}|$ is larger than the critical value $t_{N-2,0.025}$, then H_0 is rejected at the 5% level of significance, i. e. the data set $\{A_{n,k},\, n = 1, \ldots, N_k$ and $k = 1, \ldots, K\}$ comes from a common AP. The derivation of (49.31) is given in the Appendix.

To test whether the data comes from an RP or a GP, H_0 becomes $\ln r = 0$ or its equivalence $r = 1$ and H_1 becomes $\ln r \neq 0$ or its equivalence $r \neq 1$, and the t-test statistic is obtained simply by replacing t_{A}, $\hat{d}$ and $\hat{\sigma}^2_{\mathrm{A},\varepsilon}$ with t_{G}, $\ln \hat{r}$ and $\hat{\sigma}^2_{\mathrm{G},\varepsilon}$ in (49.31).

One point worth noting is that, for testing purposes, each $\varepsilon_{\mathrm{A},n,k}$ (or $\varepsilon_{\mathrm{G},n,k}$) is essentially normally distributed, e.g. see *Gujarati* [49.32], p. 282. It is difficult to evaluate the normality assumption for a sample of only 20 observations, and formal test procedures are presented in *Ramsey* and *Ramsey* [49.33].

49.5 Estimating the Means and Variances

The work in this section is substantially based on *Leung* [49.4, 11].

49.5.1 Estimating $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ of $\bar{A}_n$s

First, the mean and variance of $\bar{A}_1$ are estimated using the relevant estimators with the formulae given below.

First we denote

$$\bar{W}_{\mathrm{A},n} \equiv \frac{\sum_{k=1}^{K^*} W_{\mathrm{A},n,k}}{K^*} \quad \text{and} \quad \mu_{\bar{A}_1} \equiv \frac{\sum_{k=1}^{K^*} \mu_{A_1,k}}{K^*}\,.$$

From Definition 49.1, $W_{\mathrm{A},n,k}$s are i.i.d., we have

$$\mathrm{E}(\bar{W}_{\mathrm{A},n}) \equiv \frac{\sum_{k=1}^{K^*}\mathrm{E}(W_{\mathrm{A},n,k})}{K^*} = \frac{\sum_{k=1}^{K^*}\mu_{A_1,k}}{K^*} \equiv \mu_{\bar{A}_1}$$

and

$$\mathrm{V}(\bar{W}_{\mathrm{A},n}) \equiv \frac{\sum_{k=1}^{K^*}\mathrm{V}(W_{\mathrm{A},n,k})}{(K^*)^2} = \frac{\sum_{k=1}^{K^*}\sigma^2_{A_1,k}}{(K^*)^2} \equiv \frac{\sigma^2_{\bar{A}_1}}{K^*}\,.$$

From (49.14), (49.15), (49.16) and (49.17), we obtain

$$\mathrm{E}(\bar{W}_{\mathrm{A},n}) \equiv \frac{\sum_{k=1}^{K^*} \mathrm{E}(W_{\mathrm{A},n,k})}{K^*} = \frac{K^*\alpha_{\mathrm{A}}}{K^*} = \alpha_{\mathrm{A}}$$

and

$$\begin{aligned}\mathrm{V}(\bar{W}_{\mathrm{A},n}) &\equiv \frac{\sum_{k=1}^{K^*} \mathrm{V}(W_{\mathrm{A},n,k})}{(K^*)^2} = \frac{\sum_{k=1}^{K^*} \mathrm{V}(\alpha_A + \varepsilon_{\mathrm{A},n,k})}{(K^*)^2} \\ &= \frac{\sum_{k=1}^{K^*} \mathrm{V}(\varepsilon_{\mathrm{A},n,k})}{(K^*)^2} = \frac{\sum_{k=1}^{K^*} \sigma^2_{\mathrm{A},\varepsilon}}{(K^*)^2} \\ &= \frac{K^*\sigma^2_{\mathrm{A},\varepsilon}}{(K^*)^2} = \frac{\sigma^2_{\mathrm{A},\varepsilon}}{K^*} .\end{aligned}$$

Therefore, the first estimators for $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ are denoted and given by

$$\hat{\mu}_{\bar{A}_1,1} = \hat{\alpha}_{\mathrm{A}} \tag{49.32}$$

and

$$\hat{\sigma}^2_{\bar{A}_1,1} = \hat{\sigma}^2_{\mathrm{A},\varepsilon} . \tag{49.33}$$

Alternatively, since $W_{\mathrm{A},n,k}$s are i.i.d. with mean $\mu_{W_{\mathrm{A},n,k}} = \mu_{A_1,k}$ and variance $\sigma^2_{W_{\mathrm{A},n,k}} = \sigma^2_{A_1,k}$, it is plausible to estimate $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ by the sample mean and sample variance of $\hat{W}_{\mathrm{A},n,k}$s, where $\hat{W}_{\mathrm{A},n,k} = A_{n,k} + (n-1)\hat{d}$. Hence, the second estimators for $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ are denoted and given by

$$\begin{aligned}\hat{\mu}_{\bar{A}_1,2} &= \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} \hat{W}_{\mathrm{A},n,k}}{N} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} [A_{n,k} + (n-1)\hat{d}]}{N} \\ &= \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k}}{N} + \frac{\hat{d}\left(\sum_{k=1}^{K} N_k^2 - N\right)}{2N}\end{aligned} \tag{49.34}$$

and

$$\begin{aligned}\hat{\sigma}^2_{\bar{A}_1,2} = &\frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} \left[A_{n,k} + (n-1)\hat{d}\right]^2}{N-1} \\ &- \frac{\left\{\sum_{k=1}^{K}\sum_{n=1}^{N_k} [A_{n,k} + (n-1)\hat{d}]\right\}^2}{N(N-1)} .\end{aligned} \tag{49.35}$$

Notice that the second estimator $\hat{\mu}_{\bar{A}_1,2}$ given by (49.34) is the same as the first estimator $\hat{\mu}_{\bar{A}_1,1}$ given by (49.32) or (49.28).

It is also plausible to obtain the third estimator for $\mu_{\bar{A}_1}$ provided $N_k \cong N_0$ for $k = 1, \ldots, K$ as follows.

Let $S_N = \sum_{k=1}^{K}\sum_{n=1}^{N_k} A_{n,k} = \sum_{k=1}^{K}\sum_{n=1}^{N_k} [A_{1,k} - (n-1)d]$. Then

$$\begin{aligned}\mathrm{E}(S_N) &= \sum_{k=1}^{K} N_k \mathrm{E}(A_{1,k}) - \frac{d\left(\sum_{k=1}^{K} N_k^2 - N\right)}{2} \\ &= \sum_{k=1}^{K} N_k \mu_{A_{1,k}} - \frac{d\left(\sum_{k=1}^{K} N_k^2 - N\right)}{2} .\end{aligned}$$

If $N_k \cong N_0$ for $k = 1, \ldots, K$; then

$$\begin{aligned}\mathrm{E}(S_N) &\cong N_0 \sum_{k=1}^{K} \mu_{A_1,k} - \frac{dK\left(N_0^2 - N_0\right)}{2} \\ &= KN_0\mu_{\bar{A}_1} - \frac{dKN_0(N_0-1)}{2} .\end{aligned}$$

After transposition, we have

$$\mu_{\bar{A}_1} \cong \frac{\mathrm{E}(S_N)}{KN_0} + \frac{d(N_0-1)}{2} .$$

Hence, the third estimator for $\mu_{\bar{A}_1}$, provided $N_k \cong N_0$, for $k = 1, \ldots, K$, is denoted and given by

$$\begin{aligned}\hat{\mu}_{\bar{A}_1,3} &\cong \frac{S_N}{KN_0} + \frac{\hat{d}(N_0-1)}{2} \\ &= \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_0} A_{n,k}}{KN_0} + \frac{\hat{d}(N_0-1)}{2} .\end{aligned} \tag{49.36}$$

In fact, we can deduce (49.36) directly from (49.28) by putting $N_k \cong N_0$ and $N \cong KN_0$. In other words, the third estimator for $\mu_{\bar{A}_1}$ is indeed the first estimator but calculated approximately using (49.36). As a whole, only one estimator for $\mu_{\bar{A}_1}$, namely $\hat{\mu}_{\bar{A}_1,1}$ has been derived so far. It is furthermore plausible to obtain the second (*fourth*) and third (*fifth*) estimators for $\mu_{\bar{A}_1}$, provided $N_k \cong N_0$ for $k = 1, \ldots, K$, as follows.

In view of the fact that $\mathrm{E}(\bar{W}_{\mathrm{A},n}) = \mu_{\bar{A}_1}$, we can write

$$\bar{W}_{\mathrm{A},n} = \mu_{\bar{A}_1}(1 + \delta_{\mathrm{A},n}) . \tag{49.37}$$

1. We have

$$\mathrm{E}\left(\frac{\bar{W}_{\mathrm{A},n}}{\mu_{\bar{A}_1}}\right) = 1 + \mathrm{E}(\delta_{\mathrm{A},n})$$

and so it follows that

$$\mathrm{E}(\delta_{\mathrm{A},n}) = 0\,. \tag{49.38}$$

2. We obtain

$$\mathrm{V}\left(\frac{\bar{W}_{\mathrm{A},n}}{\mu_{\bar{A}_1}}\right) = \mathrm{V}(1+\delta_{\mathrm{A},n})\,,$$

$$\frac{\mathrm{V}(\bar{W}_{\mathrm{A},n})}{\mu^2_{\bar{A}_1}} = \mathrm{V}(\delta_{\mathrm{A},n})$$

and so it follows that

$$\mathrm{V}(\delta_{\mathrm{A},n}) = \frac{\sigma^2_{\bar{A}_1}}{K^*\mu^2_{\bar{A}_1}}\,. \tag{49.39}$$

3. Taking the logarithm of $\bar{W}_{\mathrm{A},n} \equiv \frac{\sum_{k=1}^{K^*} W_{\mathrm{A},n,k}}{K^*}$, using equation (49.14) and the fact that $\varepsilon_{\mathrm{A},n,k} \equiv \varepsilon_{\mathrm{A},n}$, irrespective of k, and taking the logarithm for (49.37), we obtain

$$\ln \bar{W}_{\mathrm{A},n} = \ln\left[\alpha_{\mathrm{A}}\left(1+\frac{\varepsilon_{\mathrm{A},n}}{\alpha_{\mathrm{A}}}\right)\right] = \ln\alpha_{\mathrm{A}} + \ln\left(1+\frac{\varepsilon_{\mathrm{A},n}}{\alpha_{\mathrm{A}}}\right) \tag{49.40}$$

and

$$\ln \bar{W}_{\mathrm{A},n} = \ln\mu_{\bar{A}_1} + \ln(1+\delta_{\mathrm{A},n})\,. \tag{49.41}$$

Taking the expectations of (49.40) and (49.41), equating them, and expanding the logarithm series, we have

$$\begin{aligned}
&\ln\alpha_{\mathrm{A}} + \mathrm{E}\left(\frac{\varepsilon_{\mathrm{A},n}}{\alpha_{\mathrm{A}}} - \frac{\varepsilon^2_{\mathrm{A},n}}{2\alpha^2_{\mathrm{A}}} + \frac{\varepsilon^3_{\mathrm{A},n}}{3\alpha^3_{\mathrm{A}}} - \cdots\right) \\
&= \ln\mu_{\bar{A}_1} + \mathrm{E}\left(\delta_{\mathrm{A},n} - \frac{\delta^2_{\mathrm{A},n}}{2} + \frac{\delta^3_{\mathrm{A},n}}{3} - \cdots\right)\,, \\
&\ln\alpha_{\mathrm{A}} + \frac{1}{\alpha_{\mathrm{A}}}\mathrm{E}(\varepsilon_{\mathrm{A},n}) - \frac{1}{2\alpha^2_{\mathrm{A}}}\mathrm{E}(\varepsilon^2_{\mathrm{A},n}) \\
&\cong \ln\mu_{\bar{A}_1} + \mathrm{E}(\delta_{\mathrm{A},n}) - \frac{1}{2}\mathrm{E}(\delta^2_{\mathrm{A},n})\,, \\
&\ln\alpha_{\mathrm{A}} - \frac{1}{2\alpha^2_{\mathrm{A}}}\mathrm{V}(\varepsilon_{\mathrm{A},n}) \\
&= \ln\mu_{\bar{A}_1} - \frac{1}{2}\mathrm{V}(\delta_{\mathrm{A},n}) \quad \text{by (49.16) and (49.38)} \\
&\ln\alpha_{\mathrm{A}} - \frac{\sigma^2_{\mathrm{A},\varepsilon}}{2\alpha^2_{\mathrm{A}}} \\
&= \ln\mu_{\bar{A}_1} - \frac{\sigma^2_{\bar{A}_1}}{2K^*\mu^2_{\bar{A}_1}} \quad \text{by (49.17) and (49.39);}
\end{aligned}$$

4. $\mu_{\bar{A}_1}$ must satisfy the equation

$$\ln\left(\frac{\mu_{\bar{A}_1}}{\alpha_{\mathrm{A}}}\right) - \frac{1}{2}\left(\frac{\sigma^2_{\bar{A}_1}}{K^*\mu^2_{\bar{A}_1}} - \frac{\sigma^2_{\mathrm{A},\varepsilon}}{\alpha^2_{\mathrm{A}}}\right) = 0\,.$$

5. We can estimate $\mu_{\bar{A}_1}$, provided $N_k \cong N_0$ for $k = 1, \ldots, K$, i. e., $K = K^*$, by $\hat{\mu}_{\bar{A}_1,2}$, which satisfies the equation

$$\ln\left(\frac{\mu_{\bar{A}_1}}{\hat{\alpha}_{\mathrm{A}}}\right) - \frac{1}{2}\left(\frac{\hat{\sigma}^2_{\bar{A}_1,1}}{K\mu^2_{\bar{A}_1}} - \frac{\hat{\sigma}^2_{\mathrm{A},\varepsilon}}{\hat{\alpha}^2_{\mathrm{A}}}\right) = 0 \tag{49.42}$$

or by $\hat{\mu}_{\bar{A}_1,3}$, which satisfies the equation

$$\ln\left(\frac{\mu_{\bar{A}_1}}{\hat{\alpha}_{\mathrm{A}}}\right) - \frac{1}{2}\left(\frac{\hat{\sigma}^2_{\bar{A}_1,2}}{K\mu^2_{\bar{A}_1}} - \frac{\hat{\sigma}^2_{\mathrm{A},\varepsilon}}{\hat{\alpha}^2_{\mathrm{A}}}\right) = 0\,, \tag{49.43}$$

where $\hat{\alpha}_{\mathrm{A}}, \hat{\sigma}^2_{\mathrm{A},\varepsilon}, \hat{\sigma}^2_{\bar{A}_1,1}$ and $\hat{\sigma}^2_{\bar{A}_1,2}$ are given by (49.28), (49.29), (49.33) and (49.35), respectively.

Clearly, if $d = 0$, the parameters $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ can be estimated using the sample mean and sample variance, which are given by

$$\hat{\mu}_{A_1,4} = \frac{\sum_{n=1}^{N} A_n}{N} \quad \text{and} \quad \hat{\sigma}^2_{A_1,3} = \frac{\sum_{n=1}^{N}(A_n - \hat{\mu}_{A_1,4})^2}{N-1}\,. \tag{49.44}$$

Secondly, we use (49.3) and (49.4), and let $N_1 \geq N_2 \geq \ldots \geq N_K$, the means and variances of $\bar{A}_n$ for $n = 2, 3, \ldots, N_k$ and $k = 1, \ldots, K$ are estimated using the following formulae:

$$\begin{aligned}
&\hat{\mu}_{\bar{A}_n} = \hat{\mu}_{\bar{A}_1} - (n-1)\hat{d} \quad \text{and} \\
&\hat{\sigma}^2_{\bar{A}_n} = \frac{\hat{\sigma}^2_{\bar{A}_1}}{K^*} \quad \text{for } n = 2, 3, \ldots, N_1\,.
\end{aligned} \tag{49.45}$$

49.5.2 Estimating $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$ of $\bar{G}_n$s

First, the mean and variance of $\bar{G}_1$ are estimated using the relevant estimators with the formulae given below.

First we denote

$$\bar{W}_{\mathrm{G},n} \equiv \frac{\sum_{k=1}^{K^*} W_{\mathrm{G},n,k}}{K^*}\,.$$

From Definition 49.2, $W_{G,n,k}$s are i.i.d., we have

$$\mathrm{E}(\bar{W}_{\mathrm{G},n}) \equiv \frac{\sum_{k=1}^{K^*} \mathrm{E}(W_{\mathrm{G},n,k})}{K^*} = \frac{\sum_{k=1}^{K^*} \mu_{G_1,k}}{K^*} \equiv \mu_{\bar{G}_1}$$

and

$$\mathrm{V}(\bar{W}_{\mathrm{G},n}) \equiv \frac{\sum_{k=1}^{K^*} \mathrm{V}(W_{\mathrm{G},n,k})}{(K^*)^2} = \frac{\sum_{k=1}^{K^*} \sigma^2_{G_1,k}}{(K^*)^2} \equiv \frac{\sigma^2_{\bar{G}_1}}{K^*} .$$

From (49.22), (49.24) and (49.25), we obtain

$$\begin{aligned}
\mathrm{E}(\bar{W}_{\mathrm{G},n}) &\equiv \frac{\sum_{k=1}^{K^*} \mathrm{E}(W_{\mathrm{G},n,k})}{K^*} = \frac{\sum_{k=1}^{K^*} \mathrm{E}(\mathrm{e}^{\alpha_{\mathrm{G}}+\varepsilon_{\mathrm{G},n,k}})}{K^*} \\
&= \frac{\mathrm{e}^{\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \mathrm{E}(\mathrm{e}^{\varepsilon_{\mathrm{G},n,k}})}{K^*} \\
&= \frac{\mathrm{e}^{\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \mathrm{E}\left(1+\varepsilon_{\mathrm{G},n,k}+\frac{\varepsilon^2_{\mathrm{G},n,k}}{2!}+\cdots\right)}{K^*} \\
&\cong \frac{\mathrm{e}^{\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \left[1+\mathrm{E}(\varepsilon_{\mathrm{G},n,k})+\frac{1}{2}\mathrm{E}(\varepsilon^2_{\mathrm{G},n,k})\right]}{K^*} \\
&= \frac{\mathrm{e}^{\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \left[1+\frac{1}{2}\mathrm{V}(\varepsilon_{\mathrm{G},n,k})\right]}{K^*} \\
&= \mathrm{e}^{\alpha_{\mathrm{G}}}\left(1+\frac{\sigma^2_{\mathrm{G},\varepsilon}}{2}\right) .
\end{aligned}$$

From (49.22) and (49.25), we obtain

$$\begin{aligned}
\mathrm{V}(\bar{W}_{\mathrm{G},n}) &\equiv \frac{\sum_{k=1}^{K^*} \mathrm{V}(W_{\mathrm{G},n,k})}{(K^*)^2} = \frac{\sum_{k=1}^{K^*} \mathrm{V}\left(\mathrm{e}^{\alpha_{\mathrm{G}}+\varepsilon_{\mathrm{G},n,k}}\right)}{(K^*)^2} \\
&= \frac{\mathrm{e}^{2\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \mathrm{V}\left(\mathrm{e}^{\varepsilon_{\mathrm{G},n,k}}\right)}{(K^*)^2} \\
&= \frac{\mathrm{e}^{2\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \mathrm{V}\left(1+\varepsilon_{\mathrm{G},n,k}+\frac{\varepsilon^2_{\mathrm{G},n,k}}{2!}+\cdots\right)}{(K^*)^2} \\
&\cong \frac{\mathrm{e}^{2\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \mathrm{V}(1+\varepsilon_{\mathrm{G},n,k})}{(K^*)^2} \\
&= \frac{\mathrm{e}^{2\alpha_{\mathrm{G}}} \sum_{k=1}^{K^*} \mathrm{V}(\varepsilon_{\mathrm{G},n,k})}{(K^*)^2} = \frac{\mathrm{e}^{2\alpha_{\mathrm{G}}}\sigma^2_{\mathrm{G},\varepsilon}}{K^*} .
\end{aligned}$$

Therefore, the first estimators for $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$ are denoted and given by

$$\hat{\mu}_{\bar{G}_1,1} = \mathrm{e}^{\hat{\alpha}_{\mathrm{G}}}\left(1+\frac{\hat{\sigma}^2_{\mathrm{G},\varepsilon}}{2}\right) \tag{49.46}$$

and

$$\hat{\sigma}^2_{\bar{G}_1,1} = \mathrm{e}^{2\hat{\alpha}_{\mathrm{G}}}\hat{\sigma}^2_{\mathrm{G},\varepsilon} . \tag{49.47}$$

Alternatively, since $W_{\mathrm{G},n,k}$s are i.i.d. with mean $\mu_{W_{\mathrm{G},n,k}} = \mu_{G_1,k}$ and variance $\sigma^2_{W_{\mathrm{G},n,k}} = \sigma^2_{G_1,k}$, it is plausible for us to estimate $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$ by the sample mean and sample variance of $\hat{W}_{\mathrm{G},n,k}$s, where $\hat{W}_{\mathrm{G},n,k} = \hat{r}^{(n-1)}G_{n,k}$. Hence, the second estimators for $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$ are denoted and given by

$$\hat{\mu}_{\bar{G}_1,2} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} \hat{W}_{\mathrm{G},n,k}}{N} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} (\hat{r}^{(n-1)}G_{n,k})}{N} \tag{49.48}$$

and

$$\hat{\sigma}^2_{\bar{G}_1,2} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} (\hat{r}^{(n-1)}G_{n,k})^2 - \frac{\left[\sum_{k=1}^{K}\sum_{n=1}^{N_k} (\hat{r}^{(n-1)}G_{n,k})\right]^2}{N}}{N-1} . \tag{49.49}$$

It is also plausible for us to obtain the third estimator for $\mu_{\bar{G}_1}$ provided $N_k \cong N_0$ for $k = 1, \ldots, K$.

Let

$$\begin{aligned}
S_N &= \sum_{k=1}^{K}\sum_{n=1}^{N_k} G_{n,k} = \sum_{k=1}^{K}\sum_{n=1}^{N_k}\left(\frac{G_{1,k}}{r^{(n-1)}}\right) \\
&= \sum_{k=1}^{K} G_{1,k} \sum_{n=1}^{N_k} \frac{1}{r^{(n-1)}} \\
&= \frac{1}{1-r^{-1}} \sum_{k=1}^{K}\left(1-r^{-N_k}\right) G_{1,k} .
\end{aligned}$$

Then

$$\mathrm{E}(S_N) = \frac{1}{1-r^{-1}} \sum_{k=1}^{K}\left(1-r^{-N_k}\right)\mu_{G_1,k} .$$

If $N_k \cong N_0$ for $k = 1, \ldots, K$, then

$$\mathrm{E}(S_N) \cong \frac{K\mu_{\bar{G}_1}\left(1-r^{-N_0}\right)}{1-r^{-1}} .$$

After transposition, we have

$$\mu_{\tilde{G}_1} \cong \frac{\mathrm{E}(S_N)\left(1-r^{-1}\right)}{K(1-r^{-N_0})} .$$

Hence, the third estimator for $\mu_{\tilde{G}_1}$, provided $N_k \cong N_0$, for $k = 1, \ldots, K$, is denoted and given by

$$\hat{\mu}_{\tilde{G}_1,3} \cong \frac{S_N\left(1-\hat{r}^{-1}\right)}{K(1-\hat{r}^{-N_0})} = \frac{\left(1-\hat{r}^{-1}\right)\sum_{k=1}^{K}\sum_{n=1}^{N_0} G_{n,k}}{K(1-\hat{r}^{-N_0})} . \tag{49.50}$$

It is furthermore plausible for us to obtain the fourth and fifth estimators for $\mu_{\tilde{G}_1}$ provided $N_k \cong N_0$ for $k = 1, \ldots, K$, as follows:

Since $\mathrm{E}(\bar{W}_{\mathrm{G},n}) = \mu_{\tilde{G}_1}$, we can write

$$\bar{W}_{\mathrm{G},n} = \mu_{\tilde{G}_1}(1+\delta_{\mathrm{G},n}) . \tag{49.51}$$

1. We have

$$\mathrm{E}\left(\frac{\bar{W}_{\mathrm{G},n}}{\mu_{\tilde{G}_1}}\right) = 1+\mathrm{E}(\delta_{\mathrm{G},n})$$

and so it follows that

$$\mathrm{E}(\delta_{\mathrm{G},n}) = 0 . \tag{49.52}$$

2. We obtain

$$\mathrm{V}\left(\frac{\bar{W}_{\mathrm{G},n}}{\mu_{\tilde{G}_1}}\right) = \mathrm{V}(1+\delta_{\mathrm{G},n}) ,$$

$$\frac{\mathrm{V}(\bar{W}_{\mathrm{G},n})}{\mu^2_{\tilde{G}_1}} = \mathrm{V}(\delta_{\mathrm{G},n})$$

and so it follows that

$$\mathrm{V}(\delta_{\mathrm{G},n}) = \frac{\sigma^2_{\tilde{G}_1}}{K^*\mu^2_{\tilde{G}_1}} . \tag{49.53}$$

3. Taking the logarithm of $\bar{W}_{\mathrm{G},n} \equiv \frac{\sum_{k=1}^{K^*} W_{\mathrm{G},n,k}}{K^*}$, using (49.21) and the fact that $\varepsilon_{\mathrm{G},n,k} \equiv \varepsilon_{\mathrm{G},n}$, irrespective of k, and taking the logarithm of (49.51), we obtain

$$\ln \bar{W}_{\mathrm{G},n} = \alpha_{\mathrm{G}} + \varepsilon_{\mathrm{G},n} \tag{49.54}$$

and

$$\ln \bar{W}_{\mathrm{G},n} = \ln \mu_{\tilde{G}_1} + \ln(1+\delta_{\mathrm{G},n}) \tag{49.55}$$

Taking the expectations of (49.54) and (49.55), equating them, and expanding the logarithm series, we have

$$\alpha_{\mathrm{G}} + \mathrm{E}(\varepsilon_{\mathrm{G},n}) = \ln \mu_{\tilde{G}_1} + \mathrm{E}\left(\delta_{\mathrm{G},n} - \frac{\delta^2_{\mathrm{G},n}}{2} + \frac{\delta^3_{\mathrm{G},n}}{3} - \cdots\right) ,$$

$$\alpha_{\mathrm{G}} \cong \ln \mu_{\tilde{G}_1} + \mathrm{E}(\delta_{\mathrm{G},n}) - \frac{1}{2}\mathrm{E}(\delta^2_{\mathrm{G},n})$$

$$= \ln \mu_{\tilde{G}_1} - \frac{1}{2}\mathrm{V}(\delta_{\mathrm{G},n})$$

$$= \ln \mu_{\tilde{G}_1} - \frac{\sigma^2_{\tilde{G}_1}}{2K^*\mu^2_{\tilde{G}_1}} \quad \text{by (49.53)} .$$

4. $\mu_{\tilde{G}_1}$ must satisfy the equation

$$2K^*(\ln \mu_{\tilde{G}_1} - \alpha_{\mathrm{G}})\mu^2_{\tilde{G}_1} - \sigma^2_{\tilde{G}_1} = 0 .$$

5. We can estimate $\mu_{\tilde{G}_1}$, provided $N_k \cong N_0$ for $k = 1, \ldots, K$, i. e., $K = K^*$, by $\hat{\mu}_{\tilde{G}_1,4}$ which satisfies the equation

$$2K(\ln \mu_{\tilde{G}_1} - \hat{\alpha}_{\mathrm{G}})\mu^2_{\tilde{G}_1} - \hat{\sigma}^2_{\tilde{G}_1,1} = 0 \tag{49.56}$$

or by $\hat{\mu}_{\tilde{G}_1,5}$ which satisfies the equation

$$2K(\ln \mu_{\tilde{G}_1} - \hat{\alpha}_{\mathrm{G}})\mu^2_{\tilde{G}_1} - \hat{\sigma}^2_{\tilde{G}_1,2} = 0 , \tag{49.57}$$

where $\hat{\alpha}_{\mathrm{G}}$, $\hat{\sigma}^2_{\mathrm{G},\varepsilon}$, $\hat{\sigma}^2_{\tilde{G}_1,1}$ and $\hat{\sigma}^2_{\tilde{G}_1,2}$ are given by (49.28), (49.29) (where $\hat{d}$, $A_{n,k}$ are replaced by $\ln \hat{r}$ and $\ln G_{n,k}$), (49.47) and (49.49), respectively.

Clearly, if $\ln r = 0$ or $r = 1$, the parameters $\mu_{\tilde{G}_1}$ and $\sigma^2_{\tilde{G}_1}$ can be estimated using the sample mean and sample variance, which are given by

$$\hat{\mu}_{G_1,6} = \frac{\sum_{n=1}^{N} G_n}{N} \quad \text{and} \quad \hat{\sigma}^2_{G_1,3} = \frac{\sum_{n=1}^{N}(G_n - \hat{\mu}_{G_1,6})^2}{N-1} . \tag{49.58}$$

Secondly, we use (49.7) and (49.8), and let $N_1 \geq N_2 \geq \ldots \geq N_K$, the means and variances of $\bar{G}_n$ for $n = 2, 3, \ldots, N_k$ and $k = 1, \ldots, K$ are estimated using the following formulae:

$$\hat{\mu}_{\bar{G}_n} = \frac{\hat{\mu}_{\tilde{G}_1}}{\hat{r}^{(n-1)}} \quad \text{and} \quad \hat{\sigma}^2_{\bar{G}_n} = \frac{\hat{\sigma}^2_{\tilde{G}_1}}{\hat{r}^{2(n-1)}K^*} \quad \text{for } n = 2, 3, \ldots, N_1 . \tag{49.59}$$

49.5.3 Estimating the Means and Variances of a Single AP or GP

When $K = 1$, (49.30) becomes $N = N_1$. Note again that

1. for a single AP or GP, we simply use N to represent the number of successive events, and
2. the results listed in the next two subsections are consistent with those derived in *Leung* [49.8] and *Lam* [49.15].

A Single AP

First, the mean and variance of A_1 are estimated using the relevant estimators with the formulae given below.

The first estimators for μ_{A_1} and $\sigma^2_{A_1}$ are denoted and given by

$$\hat{\mu}_{A_1,1} = \hat{\alpha}_{\mathrm{A}} \tag{49.32.1}$$

and

$$\hat{\sigma}^2_{A_1,1} = \hat{\sigma}^2_{\mathrm{A},\varepsilon}\,, \tag{49.33.1}$$

where $\hat{\alpha}_{\mathrm{A}}$ and $\hat{\sigma}^2_{\mathrm{A},\varepsilon}$ are given by (49.28.1) and (49.29.1).

The second estimator for $\sigma^2_{A_1}$ is denoted and given by

$$\hat{\sigma}^2_{A_1,2} = \frac{\sum_{n=1}^{N}[A_n+(n-1)\hat{d}]^2 - \frac{\left\{\sum_{n=1}^{N}[A_n+(n-1)\hat{d}]\right\}^2}{N}}{N-1}\,, \tag{49.35.1}$$

where $\hat{d}$ is given by (49.27.1).

The second $\hat{\mu}_{A_1,2}$ and third $\hat{\mu}_{A_1,3}$ estimators for μ_{A_1}, respectively, satisfy the equations

$$\ln\left(\frac{\mu_{A_1}}{\hat{\alpha}_{\mathrm{A}}}\right) - \frac{1}{2}\left(\frac{\hat{\sigma}^2_{A_1,1}}{\mu^2_{A_1}} - \frac{\hat{\sigma}^2_{\mathrm{A},\varepsilon}}{\hat{\alpha}^2_{\mathrm{A}}}\right) = 0 \tag{49.42.1}$$

and

$$\ln\left(\frac{\mu_{A_1}}{\hat{\alpha}_{\mathrm{A}}}\right) - \frac{1}{2}\left(\frac{\hat{\sigma}^2_{A_1,2}}{\mu^2_{A_1}} - \frac{\hat{\sigma}^2_{\mathrm{A},\varepsilon}}{\hat{\alpha}^2_{\mathrm{A}}}\right) = 0\,, \tag{49.43.1}$$

where $\hat{\alpha}_{\mathrm{A}}$, $\hat{\sigma}^2_{\mathrm{A},\varepsilon}$, $\hat{\sigma}^2_{A_1,1}$ and $\hat{\sigma}^2_{A_1,2}$ are given by (49.28.1), (49.29.1), (49.33.1) and (49.35.1), respectively.

Clearly, if $d = 0$, the parameters μ_{A_1} and $\sigma^2_{A_1}$ can be estimated using (49.44).

Secondly, using (49.3) and (49.4), the means and variances of A_n for $n = 2, 3, \ldots, N$ are estimated using the following formulae:

$$\hat{\mu}_{A_n} = \hat{\mu}_{A_1} - (n-1)\hat{d} \quad \text{and} \quad \hat{\sigma}^2_{A_n} = \hat{\sigma}^2_{A_1} \quad \text{for } n = 2, 3, \ldots, N\,. \tag{49.45.1}$$

A Single GP

First, the mean and variance of G_1 are estimated using the relevant estimators with formulae given below.

The first estimators for μ_{G_1} and $\sigma^2_{G_1}$ are denoted and given by

$$\hat{\mu}_{G_1,1} = \mathrm{e}^{\hat{\alpha}_{\mathrm{G}}}\left(1 + \frac{\hat{\sigma}^2_{\mathrm{G},\varepsilon}}{2}\right) \tag{49.46.1}$$

and

$$\hat{\sigma}^2_{G_1,1} = \mathrm{e}^{2\hat{\alpha}_{\mathrm{G}}}\hat{\sigma}^2_{\mathrm{G},\varepsilon}\,, \tag{49.47.1}$$

where $\hat{\alpha}_{\mathrm{G}}$ and $\hat{\sigma}^2_{\mathrm{G},\varepsilon}$ are given by (49.28.1) and (49.29.1) with $\hat{d}$ and A_n replaced by $\ln \hat{r}$ and $\ln G_n$.

The second estimators for μ_{G_1} and $\sigma^2_{G_1}$ are denoted and given by

$$\hat{\mu}_{G_1,2} = \frac{\sum_{n=1}^{N}\hat{r}^{(n-1)}G_n}{N} \tag{49.48.1}$$

and

$$\hat{\sigma}^2_{G_1,2} = \frac{\sum_{n=1}^{N}\left(\hat{r}^{(n-1)}G_n\right)^2 - \frac{\left(\sum_{n=1}^{N}\hat{r}^{(n-1)}G_n\right)^2}{N}}{N-1}\,, \tag{49.49.1}$$

where $\hat{r}$ is given by (49.27.1) with A_n replaced by $\ln G_n$.

The third estimator for μ_{G_1} is denoted and given by

$$\hat{\mu}_{G_1,3} = \frac{\left(1-\hat{r}^{-1}\right)\sum_{n=1}^{N}G_n}{1-\hat{r}^{-N}}\,. \tag{49.50.1}$$

The fourth $\hat{\mu}_{G_1,4}$ and fifth $\hat{\mu}_{G_1,5}$ estimators for μ_{G_1} respectively satisfy the equations

$$2(\ln\mu_{G_1} - \hat{\alpha}_{\mathrm{G}})\mu^2_{G_1} - \hat{\sigma}^2_{G_1,1} = 0 \tag{49.56.1}$$

and

$$2(\ln\mu_{G_1} - \hat{\alpha}_{\mathrm{G}})\mu^2_{G_1} - \hat{\sigma}^2_{G_1,2} = 0\,, \tag{49.57.1}$$

where $\hat{\sigma}^2_{G_1,1}$ and $\hat{\sigma}^2_{G_1,2}$ are given by (49.47.1) and (49.49.1).

Clearly, if $r = 1$, the parameters μ_{G_1} and $\sigma^2_{G_1}$ can be estimated using (49.58).

Secondly, using (49.7) and (49.8), the means and variances of A_n for $n = 2, 3, \ldots, N$ are estimated using the following formulae:

$$\hat{\mu}_{G_n} = \frac{\hat{\mu}_{G_1}}{\hat{r}^{(n-1)}} \quad \text{and} \quad \hat{\sigma}^2_{G_n} = \frac{\hat{\sigma}^2_{G_1}}{\hat{r}^{2(n-1)}} \quad \text{for } n = 2, 3, \ldots, N . \tag{49.59.1}$$

49.6 Comparison of Estimators Using Simulation

Much of the work in this section is based on *Leung* [49.7], and *Leung* and *Lai* [49.13].

49.6.1 A Single AP or GP

Some simulation studies were performed to evaluate various estimators given in Sect. 49.5.3 and to compare the different estimates of μ_{A_1} and $\sigma^2_{A_1}$ (or μ_{G_1} and $\sigma^2_{G_1}$).

For each realization $\{A_n, n = 1, \ldots, 20\}$, the estimates $\hat{\mu}_{A_1,i}, i = 1, 2, 3, 4$ are ranked using three criteria. First, if our objective is to estimate the value of μ_{A_1}, we can compute the deviation ϕ of $\hat{\mu}_{A_1,i}$ from μ_{A_1}, i. e. $\phi = \left|\hat{\mu}_{A_1,i} - \mu_{A_1}\right|$. Secondly, if our objective is to fit A_ns values only, we can calculate the mean square error (MSE) between the fitted values $\hat{A}_{n,i} = \hat{\mu}_{A_1,i} - (n-1)\hat{d}$s and observations A_ns, i. e. $\text{MSE} = \sum_{n=1}^{N} \left(\hat{A}_{n,i} - A_n\right)^2 / N$. Thirdly, if our objective is to estimate μ_{A_1} as well as fitting values of A_ns, then we can use $\Phi = \phi + \sqrt{\text{MSE}}$. Moreover, the estimates $\hat{\sigma}^2_{A_1,1}$ and $\hat{\sigma}^2_{A_1,2}$ can be compared by their standard deviations (s.d.) from $\sigma^2_{A_1}$. The recommended estimators based on the simulation studies are summarized in Table 49.1 (for more details, see *Leung* et al. [49.7]).

Similarly, for each realization $\{G_n, n = 1, \ldots, 101\}$, the estimates $\hat{\mu}_{G_1,i}, i = 1, 2, 3, 4, 5, 6$ are ranked using the aforementioned three criteria, and the estimates $\hat{\sigma}^2_{G_1,1}$ and $\hat{\sigma}^2_{G_1,2}$ can be compared by their s.d. from $\sigma^2_{G_1}$. The recommended estimators based on the simulation studies are summarized in Table 49.2 (for more details, see *Lam* [49.15]).

49.6.2 *K* Independent, Homogeneous APs or GPs

Some simulation studies were also performed to evaluate various estimators given in Sect. 49.5.1 (or Sect. 49.5.2) and to compare the different estimates of $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$ (or $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$).

For each realization $\{A_{n,k}, n = 1, \ldots, 20$ and $k = 1, \ldots, 10\}$ (or $\{G_{n,k}, n = 1, \ldots, 101$ and $k = 1, \ldots, 10\}$), the estimates $\hat{\mu}_{\bar{A}_1,i}$, $i = 1, 2, 3, 4$ (or $\hat{\mu}_{\bar{G}_1,i}, i = 1, 2, 3, 4, 5, 6$) are ranked using the first criterion, namely the deviation ϕ of $\hat{\mu}_{\theta,i}$ from μ_θ, i. e. $\phi = \left|\hat{\mu}_{\theta,i} - \mu_\theta\right|$, and the estimates $\hat{\sigma}^2_{\theta,1}$ and $\hat{\sigma}^2_{\theta,2}$

Table 49.1 Recommended estimators for μ_{A_1} and $\sigma^2_{A_1}$

d	ϕ μ_{A_1}	MSE A_n	Φ μ_{A_1} & A_n	s. d. $\sigma^2_{A_1}$
$= 0$	$\hat{\mu}_{A_1,4}$	$\hat{\mu}_{A_1,2}$ or $\hat{\mu}_{A_1,4}$	$\hat{\mu}_{A_1,3}$ or $\hat{\mu}_{A_1,4}$	$\hat{\sigma}^2_{A_1,3}$
< 0	$\hat{\mu}_{A_1,1}$ or $\hat{\mu}_{A_1,2}$	$\hat{\mu}_{A_1,2}$	$\hat{\mu}_{A_1,2}$	$\hat{\sigma}^2_{A_1,2}$
$\in \left(0, \frac{\mu_{A_1}}{n-1}\right]$	$\hat{\mu}_{A_1,2}$	$\hat{\mu}_{A_1,2}$	$\hat{\mu}_{A_1,2}$	$\hat{\sigma}^2_{A_1,2}$

Table 49.2 Recommended estimators for μ_{G_1} and $\sigma^2_{G_1}$

r	ϕ μ_{G_1}	MSE G_n	Φ μ_{G_1} & G_n	s. d. $\sigma^2_{G_1}$
$= 1$	$\hat{\mu}_{G_1,6}$	$\hat{\mu}_{G_1,3}$ or $\hat{\mu}_{G_1,6}$	$\hat{\mu}_{G_1,6}$	$\hat{\sigma}^2_{G_1,3}$
$\in (0, 1)$	$\hat{\mu}_{G_1,5}$	$\hat{\mu}_{G_1,3}$	$\hat{\mu}_{G_1,3}$	$\hat{\sigma}^2_{G_1,2}$
> 1	$\hat{\mu}_{G_1,3}$	$\hat{\mu}_{G_1,3}$	$\hat{\mu}_{G_1,3}$	$\hat{\sigma}^2_{G_1,2}$

Table 49.3 Recommended estimators for $\mu_{\bar{A}_1}$ and $\sigma^2_{\bar{A}_1}$, and $\mu_{\bar{G}_1}$ and $\sigma^2_{\bar{G}_1}$

d	ϕ $\mu_{\bar{A}_1}$	s.d. $\sigma^2_{\bar{A}_1}$	r	ϕ $\mu_{\bar{G}_1}$	s.d. $\sigma^2_{\bar{G}_1}$
$=0$	$\hat{\mu}_{A_1,4}$	$\hat{\sigma}^2_{A_1,3}$	$r=1$	$\hat{\mu}_{G_1,6}$	$\hat{\sigma}^2_{\bar{G}_1,3}$
<0	$\hat{\mu}_{\bar{A}_1,1}$	$\hat{\sigma}^2_{\bar{A}_1,1}$ or $\hat{\sigma}^2_{\bar{A}_1,2}$	$\in(0,1)$	$\hat{\mu}_{\bar{G}_1,2}$	$\hat{\sigma}^2_{\bar{G}_1,2}$
$\in\left(0, \frac{\mu_{\bar{A}_1}}{n-1}\right]$	$\hat{\mu}_{\bar{A}_1,1}$	$\hat{\sigma}^2_{\bar{A}_1,1}$ or $\hat{\sigma}^2_{\bar{A}_1,2}$	>1	$\hat{\mu}_{\bar{G}_1,2}$ or $\hat{\mu}_{\bar{G}_1,3}$	$\hat{\sigma}^2_{\bar{G}_1,2}$

can be compared by their s.d. from σ^2_θ, where $\theta =$ either $\bar{A}_1$ or $\bar{G}_1$. The recommended estimators based on the simulation studies are summarized in Table 49.3 (for more details, see *Leung* and *Lai* [49.13]).

49.6.3 Comparison Between Averages of Estimates and Pooled Estimates

Having obtained the estimates $\hat{d}$, $\hat{\mu}_{A_1}$ and $\hat{\sigma}^2_{A_1}$ (or $\hat{r}$, $\hat{\mu}_{G_1}$ and $\hat{\sigma}^2_{G_1}$) using the relevant estimators suggested in Table 49.1 (or Table 49.2), of the parameters d, μ_{A_1} and $\sigma^2_{A_1}$ (or r, μ_{G_1} and $\sigma^2_{G_1}$) of a single system, we can compute the averages of the respective estimates for a collection of homogeneous systems and then use these averages to estimate $\hat{\mu}_{A_n}$ and $\hat{\sigma}^2_{A_n}$ (or $\hat{\mu}_{G_n}$ and $\hat{\sigma}^2_{G_n}$). *Leung* and *Lai* [49.13] drew the conclusion that, in any cases, the pooled estimates obtained using the pooled estimators for APs or GPs suggested in Table 49.3 are better than the respective averages of estimates.

49.7 Real Data Analysis

Lam et al. [49.16] presented ten examples, each analyzing a real data set using four models:

1. the GP model with a nonparametric method,
2. the HPP model,
3. the NHPP model with PLP and
4. the NHPP model with LLP.

Example 1 examines 190 data of the intervals in days between successive coal-mining disasters in Great Britain, which have been used by a number of researchers to illustrate various techniques that can be applied to point processes; see, for example, *Cox* and *Lewis* [49.17], pp. 42–43. The data set can be found in *Hand* et al. [49.34], p. 155 or *Andrews* and *Herzberz* [49.35], pp. 51–56, in which the data are recorded in more detail.

Examples 2–4 study 29, 30 and 27 data of the intervals in operating hours between successive failures of air-conditioning equipment in aircrafts 3, 6 and 7. The 13 data sets tabulate on p. 6 of *Cox* and *Lewis* [49.17], and the data sets being examined are the largest three.

Example 5 investigates 257 failure times of a computer in unspecified units. The data are given in *Cox* and *Lewis* ([49.17], p. 11).

Example 6 examines 245 arrival times of patients at an intensive care unit in a hospital. The data are given in *Cox* and *Lewis* ([49.17], p. 14 and pp. 254–255).

Examples 7 and 8 study 71 and 56 data of the arrival times to unscheduled overhauls for the no. 3 and no. 4 main propulsion diesel engines for two submarines. The two data sets tabulate on pp. 75–76 of *Ascher* and *Feingold* [49.18].

Example 9 investigates the times that 41 successive vehicles traveling northwards along the M1 motorway in England passed a fixed point near junction 13 in Bedfordshire on Saturday 23 March 1985. The data are given in *Hand* et al. ([49.34], p. 3).

Example 10 examines 136 failure times [in central processing unit (CPU) seconds, measured in terms of execution time] of a real-time command-and-control software system. The data are given in *Hand* et al. ([49.34], p. 10–11).

Lam et al. [49.16] concluded that, on average, the GP model is the best model for fitting these ten real data sets among the four models based on the MSE criterion (as defined in Sect. 49.5.1). This is the reason why the GP model can be applied to the maintenance problems.

Furthermore, *Lam* and *Chan* [49.36] applied the GP model to fit the three real data sets in Examples 1, 7 and 8 using a parametric method with one of the log-normal, exponential, gamma and Weibull distributions. The numerical results also conclude that all three data sets can be well fitted by the GP model based on the MSE criterion.

The author is currently investigating the following:

1. The ten data sets are analyzed using the AP model with a nonparametric method and the numerical results are compared with those in *Lam* et al. [49.16].
2. The data sets in Examples 2–4 (or even the 13 data sets) and the data sets in Examples 7 and 8 are respectively pooled together to estimate the parameters using the methods suggested in Sects. 49.2–49.5 for AP and GP, and the methods used in *Leung* and *Cheng* [49.3] for HPP and NHPP. The numerical results are also compared with those in *Lam* et al. [49.16] plus those obtained using the AP model.

49.8 Optimal Replacement Policies Determined Using Arithmetico-Geometric Processes

The work in this section is substantially based on *Leung* [49.5].

49.8.1 Arithmetico-Geometric Processes

A definition of an AGP is given below.

Definition 49.3

Given a sequence of random variables $H_1, H_2, \ldots$, if for some real number d and some $r > 0$, $\{[H_n + (n-1)d]r^{(n-1)}, n = 1, 2, \ldots\}$ forms an RP, then $\{H_n,\ n = 1, 2, \ldots\}$ is an AGP. The two parameters d and r are called the common difference and the common ratio of the AGP respectively.

Three specializations of an AGP are given below.

If $r > 1$ and $d \in \left(0, \frac{\mu_{H_1}}{(n-1)r^{(n-1)}}\right]$, where $n = 2, 3, \ldots$ and μ_{H_1} is the mean of the first random variable H_1, then the AGP is called a decreasing AGP. If $d < 0$ and $0 < r < 1$, then the AGP is called an increasing AGP. If $d = 0$ and $r = 1$, then the AGP reduces to an RP.

Two immediate remarks concerning the characteristics of an AGP are as follows:

1. An AGP is the name given to a series in which the general term is the product of the general term of an AP and of a GP; we take this term to be, in general,
$$H_n = \frac{H_1}{r^{(n-1)}} - (n-1)d\ .$$
2. It is evident that, if we put $r = 1$ but d $\neq 0$, or $d = 0$ but $r \neq 1$ into the above expression, the process obtained becomes an AP, or a GP. Hence, an AGP extends and generalizes an AP or a GP.

Therefore, for a deteriorating system, it is reasonable to assume that the successive operating times of the system form a decreasing AGP, whereas the corresponding consecutive repair times constitute an increasing AGP. However, the replacement times for the system are usually stochastically the same no matter how old the used system is; hence, they will form an RP. This is the motivation behind the introduction of the AGP approach.

49.8.2 Model

Before deriving new repair–replacement models, the following assumptions are stated.

1. At the beginning, a new system is used.
2. Whenever the system fails, it can be repaired. Let X_n be the survival time after the $(n-1)$th repair, then a sequence $\{X_n, n = 1, 2, \ldots\}$ forms a decreasing AGP with parameters $d_a > 0$ and $r_a > 1$ such that $\mu_{X_1}/(n-1)r_a^{(n-1)} \geq d_a$, where $\mathrm{E}(X_1) \equiv \mu_{X_1} > 0$.
3. Let Y_n be the repair time after the nth failure, then a sequence $\{Y_n, n = 1, 2, \ldots\}$ forms an increasing AGP with parameters $d_b < 0$ and $0 < r_b < 1$, and $\mathrm{E}(Y_1) \equiv \mu_{Y_1} \geq 0$. $\mu_{Y_1} = 0$ means that the repair time is negligible.
4. A sequence $\{X_n, n = 1, 2, \ldots\}$ and a sequence $\{Y_n, n = 1, 2, \ldots\}$ are independent.
5. An average operating cost rate is c_o, an average repair cost rate is c_f, and an average revenue rate of a working system is w.
6. The system may be replaced at some time by a new and identical one. An average replacement cost rate under policy T or N is c_{RT} or c_{RN}, respectively, and an average replacement downtime under policy T or N is u_{RT} or u_{RN}, respectively. Two kinds of replacement policy are considered in this model.
   a) A replacement policy T is a policy in which we replace the system whenever the working age of the system reaches T, a continuous decision variable, see *Barlow* and *Proschan* [49.19]. The working age T of a system at time t is the

cumulative survival time by time t, i. e.

$$T = \begin{cases} t - V_n, U_n + V_n \le t < U_{n+1} + V_n \\ U_{n+1}, U_{n+1} + V_n \le t < U_{n+1} + V_{n+1} \end{cases}, \quad (49.60)$$

where $U_n = \sum_{i=1}^{n} X_i$, $V_n = \sum_{i=1}^{n} Y_i$ and $U_0 = 0$, $V_0 = 0$.

b) A replacement policy N is a policy in which we replace the system at the time of Nth failure since the last replacement, a discrete decision variable, see *Nakagawa* [49.37].

Under replacement policy T or N, the problem is to determine an optimal replacement policy T^* or N^*, respectively, such that the long-run expected loss per unit total time or per unit operation time is minimized.

49.8.3 The Long-Run Expected Loss Rate

Let T_n be the time between the $(n-1)$th replacement and the nth replacement with $T_0 = 0$, then $\{T_n, n = 1, 2, \dots\}$ forms an RP. Applying known results from renewal theory (see e.g. *Ross* [49.20], pp. 51–54), the long-run expected loss per unit time is obtained by

$$l(T) \text{ or } l(N) = \frac{\text{E(loss incurred in a cycle)}}{\text{E(length of a cycle)}}, \quad (49.61)$$

where the loss is defined as the total cost minus total revenue, and a cycle is the time between two consecutive replacements.

Under the replacement policy T, denote the length of a cycle by W, then

$$W = T + V_n \quad \text{and}$$
$$U_n < T \le U_{n+1} \quad \text{for } n = 0, 1, \dots . \quad (49.62)$$

From Definition 49.3 and Assumption 3, it follows that $\mathrm{E}(Y_n) = \frac{\mu_{Y_1}}{r_b^{(n-1)}} - (n-1)d_b$. Then

$$\mathrm{E}(W) = \mathrm{E}\left(T + \sum_{j=1}^{n} Y_j\right) = \mathrm{E}\left[T + \sum_{j=1}^{\infty} Y_j I(n \ge j)\right],$$

where I is the indicator random variable defined as

$$I(n \ge j) = \begin{cases} 1 & \text{if } n \ge j \\ 0 & \text{if } n < j \end{cases}.$$

Since $n \ge j \Leftrightarrow U_j \le T$, we have

$$\begin{aligned} \mathrm{E}(W) &= \mathrm{E}\left[T + \sum_{j=1}^{\infty} Y_j I(U_j \le T)\right] \\ &= \mathrm{E}(T) + \sum_{j=1}^{\infty} \mathrm{E}(Y_j)\mathrm{E}[I(U_j \le T)] \\ &\quad \text{as } Y_j \text{ and } U_j \text{ are independent} \\ &= T + \sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right] \\ &\quad \times [1 \times \Pr(U_j \le T) + 0 \times \Pr(U_j > T)] \\ &= T + \sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right] F_j(T), \end{aligned}$$

where F_j is the cumulative distribution function of U_j.

Thus, from (49.61) the long-run expected loss per unit total time $l(T)$ under policy T is given by

$$l(T) = \frac{(c_o - w)T + c_{RT}u_{RT}}{T + \sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right]F_j(T) + u_{RT}} + \frac{c_f\left\{\sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right]F_j(T)\right\}}{T + \sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right]F_j(T) + u_{RT}} \quad (49.63)$$

and also from (49.61) the long-run expected loss per unit total time $l(N)$ under policy N is given by

$$l(N) = \left\{(c_o - w)\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_a^{(j-1)}} - (j-1)d_a\right] + c_f\sum_{j=1}^{N-1}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right] + c_{RN}u_{RN}\right\} \Big/ \left\{\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_a^{(j-1)}} - (j-1)d_a\right] + \sum_{j=1}^{N-1}\left[\frac{\mu_{Y_1}}{r_b^{(j-1)}} - (j-1)d_b\right] + u_{RN}\right\}. \quad (49.64)$$

Further, from (49.61) the long-run expected loss per unit operation time $l_{\mathrm{op}}(T)$ under policy T or $l_{\mathrm{op}}(N)$ under policy N is respectively given by

$$l_{\mathrm{op}}(T) = \frac{(c_{\mathrm{o}} - w)T + c_{RT}u_{RT}}{T} + \frac{c_{\mathrm{f}}\left\{\sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right]F_j(T)\right\}}{T} \tag{49.65}$$

or

$$l_{\mathrm{op}}(N) = \frac{(c_{\mathrm{o}} - w)\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_{\mathrm{a}}^{(j-1)}} - (j-1)d_{\mathrm{a}}\right]}{\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_{\mathrm{a}}^{(j-1)}} - (j-1)d_{\mathrm{a}}\right]} + \frac{c_{\mathrm{f}}\sum_{j=1}^{N-1}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right] + c_{RN}u_{RN}}{\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_{\mathrm{a}}^{(j-1)}} - (j-1)d_{\mathrm{a}}\right]}. \tag{49.66}$$

To simplify the optimization tasks, first operating cost is excluded ($c_{\mathrm{o}} = 0$) because this belongs to the account of production (rather than maintenance) costs; secondly, without loss of generality, w can be set equal to HK$ 1 because money can be measured in an arbitrary scale (c_{f}, C_{RT}, and C_{RN} are on the same scale of Hong Kong dollars). Hence, putting $c_{\mathrm{o}} = 0$ and $w = 1$ into (49.63) to (49.66) and then adding unity to the right-hand side of (49.63) to (49.66), we obtain the simplified versions of $l(T)$ and $l(N)$, and $l_{\mathrm{op}}(T)$ and $l_{\mathrm{op}}(N)$ namely

$$l_1(T) = \frac{(c_{\mathrm{f}} + 1)\left\{\sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right]F_j(T)\right\}}{T + \sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right]F_j(T) + u_{RT}} + \frac{(c_{RT} + 1)u_{RT}}{T + \sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right]F_j(T) + u_{RT}} \tag{49.67}$$

and

$$l_1(N) = \left\{(c_{\mathrm{f}} + 1)\sum_{j=1}^{N-1}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right] + (c_{RN} + 1)u_{RN}\right\} \Bigg/ \left\{\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_{\mathrm{a}}^{(j-1)}} - (j-1)d_{\mathrm{a}}\right] + \sum_{j=1}^{N-1}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right] + u_{RN}\right\}, \tag{49.68}$$

$$l_{\mathrm{op1}}(T) = \frac{c_{\mathrm{f}}\left\{\sum_{j=1}^{\infty}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right]F_j(T)\right\}}{T} + \frac{c_{RT}u_{RT}}{T} \tag{49.69}$$

and

$$l_{\mathrm{op1}}(N) = \frac{c_{\mathrm{f}}\sum_{j=1}^{N-1}\left[\frac{\mu_{Y_1}}{r_{\mathrm{b}}^{(j-1)}} - (j-1)d_{\mathrm{b}}\right] + c_{RN}u_{RN}}{\sum_{j=1}^{N}\left[\frac{\mu_{X_1}}{r_{\mathrm{a}}^{(j-1)}} - (j-1)d_{\mathrm{a}}\right]}, \tag{49.70}$$

respectively.

Finally, the optimal replacement policy, denoted in general by T^* or N^*, can be determined by minimizing $l(T)$, $l_{\mathrm{op}}(T)$, $l(N)$ or $l_{\mathrm{op}}(N)$ [or alternatively $l_1(T)$, $l_{\mathrm{op1}}(T)$, $l_1(N)$ or $l_{\mathrm{op1}}(N)$], respectively. Furthermore, the minimization procedure can be achieved using analytical or numerical methods.

In practice, we prefer to adopt the optimal policy N^* rather than use the optimal policy T^*, because of the much simpler form of $l(N)$, $l_{\mathrm{op}}(N)$, $l_1(N)$, and $l_{\mathrm{op1}}(N)$. Moreover, under some mild conditions, *Lam* [49.38] has proved that the optimal policy N^* is better than any policy T; in particular, it is better than the optimal policy T^*.

Note that the replacement policy that minimizes l or l_{op} also maximizes p or p_{op} because the long-run expected profit p or p_{op} per unit total time or per unit operation time is equal to the negation of the long-run expected loss l or l_{op} per unit total time or per unit operation time, respectively. The expressions for l and l_{g} include the performance measure of cost or downtime as a special case.

The expressions for c and c_{op}, the long-run expected cost per unit total time and per unit operation time, are

obtained by substituting $w = 0$ or $w = c_o = 0$ into the expressions for l and l_{op}, respectively.

The expressions for u and u_{op}, the long-run expected downtime per unit total time and per unit operation time, are obtained by substituting $w = c_o = 0$, $c_f = c_{RT} = c_{RN} = 1$ into the expressions for l and l_{op} respectively. Also, availability $= (1-u)\times 100\%$ or availability $= (1-u_{op})\times 100\%$.

49.9 Some Conclusions on the Applicability of an AP and/or a GP

As concluded in the last paragraph before Sect. 49.1 AP and/or a GP approach is considered to be relevant, realistic and direct to the modeling of deteriorating system maintenance problems. If a decreasing GP (or an increasing AP) does not fit the system's successive operating times (or repair times), a decreasing AP (or an increasing GP) may be attempted instead. Equation (49.64) with $d_b = 0$, $r_a = 1$ then becomes

$$l_{A,G}(N) = \left\{(c_o - w)\left\{\frac{N}{2}\left[2\mu_{X_1} - (N-1)d_a\right]\right\} + c_f\mu_{Y_1}\sum_{j=1}^{N-1}\frac{1}{r_b^{(j-1)}} + c_{RN}u_{RN}\right\} \Bigg/ \left\{\frac{N}{2}\left[2\mu_{X_1} - (N-1)d_a\right] + \mu_{Y_1}\sum_{j=1}^{N-1}\frac{1}{r_b^{(j-1)}} + u_{RN}\right\} \tag{49.71}$$

which is one of the four replacement models using an AP or a GP or both. This means that there are altogether four options to resolve a replacement problem when using model (49.64).

The reliability findings for the 6LXB type of engine analyzed using GPs (*Leung* and *Lee* [49.1]) and APs (*Leung* and *Kwok* [49.6]) are summarized in Table 49.4.

The reliability findings for the Benz type of gearbox analyzed using GPs (*Leung* and *Fong* [49.2]) and APs (*Leung* and *Lai* [49.10]) are summarized in Table 49.5.

In Table 49.5, the parameter $\hat{d}_b$ or $\hat{r}_b$ is larger than zero or unity for the Benz gearbox. $\hat{d}_b > 0$ or $\hat{r}_b > 1$ indicates that the repair times of the gearboxes decrease and will tend towards zero. The reasons for this phenomenon are:

1. The Kowloon Motor Bus (KMB) Company Limited spends a lot of time on the following when a gearbox first fails (see *Leemis* [49.39], p. 148)
   a) Diagnosis time: time used for fault finding, including adjustment of test equipment, carrying out checks, interpretation of information gained, verification of the conclusions drawn and deciding corrective action.
   b) Logistic time: time used in waiting for spare parts, test gears, additional tools and manpower to be transported to the system.
   c) Administrative time: time used in the allocation of repair tasks, manpower changeover due to

Table 49.4 Estimated values of common difference and ratio, and means for the 6LXB engine

Survival times	$\hat{\mu}_{X_1}$(y)	$\hat{\mu}_{X_2}$(y)	$\hat{\mu}_{X_3}$(y)	$\hat{\mu}_{X_4}$(y)
AP with $\hat{d}_a = 1.6$ y	3.4	1.8	0.2	–
GP with $\hat{r}_a = 4.533$	3.79	0.8361	0.1844	0.0407
Repair times	$\hat{\mu}_{Y_1}$(d)	$\hat{\mu}_{Y_2}$(d)	$\hat{\mu}_{Y_3}$(d)	$\hat{\mu}_{Y_4}$(d)
AP with $\hat{d}_b = -12.17$ d	8.86	21.03	33.20	–
GP with $\hat{r}_b = 0.524$	9.881	18.86	35.99	68.68

Table 49.5 Estimated values of common difference and ratio, and means for the Benz gearbox

Survival times	$\hat{\mu}_{X_1}$(y)	$\hat{\mu}_{X_2}$(y)	$\hat{\mu}_{X_3}$(y)	$\hat{\mu}_{X_4}$(y)
AP with $\hat{d}_a = 0.97$ y	3.05	2.08	1.11	0.14
GP with $\hat{r}_a = 2.004$	1.969	0.9825	0.4903	0.2447
Repair times	$\hat{\mu}_{Y_1}$(d)	$\hat{\mu}_{Y_2}$(d)	$\hat{\mu}_{Y_3}$(d)	$\hat{\mu}_{Y_4}$(d)
AP with $\hat{d}_b = 34.07$ d	85.46	51.39	17.32	–
GP with $\hat{r}_b = 2.096$	37.25	17.77	8.479	4.045

demarcation arrangements, official breaks, disputes, etc.
2. KMB gains repair experience from the first failure, which is used to improve their time management, so repair time decreases.
3. When a gearbox is taken out of a bus, there is no follow-up tracing of the gearbox and hence we are unable to find exact consecutive repair times.

The optimal replacement policy based on minimum cost is to replace the engine or gearbox after the second or third failure using the AP or GP approach. Notice that theoretically it is replaced after the ninth failure of the engine or gearbox using the GP approach; this is possible since a decreasing GP converges to zero (but a decreasing AP produces negative values, which are nonexistent in a reliability context).

Based on the four real case studies, we observe that both approaches are applicable in solving reliability problems. As to which one is more appropriate to a given set of reliability data, some criteria have to be established. Once we have criteria comparing the results using the AP and GP approaches, we can separately compare the findings obtained in *Leung* and *Lee* [49.1] with those in *Leung* and *Kwok* [49.6], and *Leung* and *Fong* [49.2] with *Leung* and *Lai* [49.10].

49.10 Concluding Remarks

There follow five notes concerning the application of the models given in the previous sections.

The first note concerns the third question: after fitting an AP (or a GP) model to the data set, how good is the fit?

Estimation of parameters is properly only a precursor to further analysis. The techniques outlined in Sects. 49.2– 49.5 may be extended to provide a basis for confidence bounds, tests for comparing different sets of event counts, and so on. *Lam* et al. [49.16] obtained the asymptotic distributions of the nonparametric estimators of r, μ_{G_1} and $\sigma^2_{G_1}$. By a parametric approach, *Lam* and *Chan* [49.36] also obtained the estimators of r, μ_{G_1} and $\sigma^2_{G_1}$ and their asymptotic distributions.

Scarf [49.40], on p. 498, has recommended that, if the assumptions of a simple AP or GP model are not

Table 49.6 Summary of useful results of both AP and GP processes

APs equation given by	GPs equation given by
$\hat{d}$ (49.27) or (49.27.1)	$\ln \hat{r}$ (49.27) or (49.27.1) with $A_{n,k}$ replaced by $\ln G_{n,k}$
$\hat{\alpha}_{\mathrm{A}}$ (49.28) or (49.28.1)	$\hat{\alpha}_{\mathrm{G}}$ (49.28) or (49.28.1) with $A_{n,k}$ replaced by $\ln G_{n,k}$
$\hat{\sigma}_{\mathrm{A},\varepsilon}$ (49.29) or (49.29.1)	$\hat{\sigma}_{\mathrm{G},\varepsilon}$ (49.29) or (49.29.1) with $A_{n,k}$ replaced by $\ln G_{n,k}$
t_{A} (49.31)	t_{G} (49.31) with $A_{n,k}$ replaced by $\ln G_{n,k}$
For $d \neq 0$	**For $r \neq 1$**
$\hat{\mu}_{\bar{A}_1,1}$ (49.32) or (49.32.1)	$\hat{\mu}_{\bar{G}_1,1}$ (49.46) or (49.46.1)
–	$\hat{\mu}_{\bar{G}_1,2}$ (49.48) or (49.48.1)
–	$\hat{\mu}_{\bar{G}_1,3}$ (49.50) or (49.50.1)
$\hat{\mu}_{\bar{A}_1,2}$ (49.42) or (49.42.1)	$\hat{\mu}_{\bar{G}_1,4}$ (49.56) or (49.56.1)
$\hat{\mu}_{\bar{A}_1,3}$ (49.43) or (49.43.1)	$\hat{\mu}_{\bar{G}_1,5}$ (49.57) or (49.57.1)
$\hat{\sigma}^2_{\bar{A}_1,1}$ (49.33) or (49.33.1)	$\hat{\sigma}^2_{\bar{G}_1,1}$ (49.47) or (49.47.1)
$\hat{\sigma}^2_{\bar{A}_1,2}$ (49.35) or (49.35.1)	$\hat{\sigma}^2_{\bar{G}_1,2}$ (49.48) or (49.48.1)
For $d = 0$	**For $r = 1$**
$\hat{\mu}_{A_1,4}, \hat{\sigma}^2_{A_1,3}$ (49.44)	$\hat{\mu}_{G_1,6}, \hat{\sigma}^2_{G_1,3}$ (49.58)
$\hat{\mu}_{\bar{A}_n}, \hat{\sigma}^2_{\bar{A}_n}$ (49.45) or (49.45.1)	$\hat{\mu}_{\bar{G}_n}, \hat{\sigma}^2_{\bar{G}_n}$ (49.59) or (49.59.1)
Replacement model	**Replacement model**
$l_{\mathrm{A}}(T)$ (49.72)	$l_{\mathrm{G}}(T)$ (49.74)
$l_{\mathrm{A}}(N)$ (49.73)	$l_{\mathrm{G}}(N)$ (49.75)

valid, and in practice this is usually so, then there are two possible routes:

1. extend the model with extra parameters, here d and r, making greater demands on the available data;
2. use the simple model to obtain a crude approximation to the optimum policy.

The second note relates to route 1. The author is focusing his efforts on developing a procedure of statistical inference for an AGP, since fitting a model to failure and/or repair data is preliminary to the utilization of an optimization model, from which an optimal maintenance policy based on minimizing loss, cost or downtime may be found. Naturally, the development of such a procedure involves much more mathematics than that for a GP by *Lam* [49.15] or for an AP by *Leung* [49.8]. Once the procedure is warranted, two parallel case studies using an AGP approach for the same set of real maintenance data of engines and gearboxes will be carried out and then findings will be compared with those obtained in *Leung* et al. [49.1, 4, 5, 10]; these case studies will be presented in two future papers.

The third note relates to route 2. Estimation of parameters is also a precursor to practical use of an optimization model. Two AP models used in resolving replacement problems are obtained by putting $r_b = 1$ in (49.63) and $r_a = r_b = 1$ in (49.64), namely the long-run expected loss per unit total time under policy T, which is given by

$$l_A(T) = \frac{(c_o - w)T + c_{RT}u_{RT}}{T + \sum_{j=1}^{\infty}[\mu_{Y_1} - (j-1)d_b]F_j(T) + u_{RT}} + \frac{c_f\left\{\sum_{j=1}^{\infty}[\mu_{Y_1} - (j-1)d_b]F_j(T)\right\}}{T + \sum_{j=1}^{\infty}[\mu_{Y_1} - (j-1)d_b]F_j(T) + u_{RT}} \tag{49.72}$$

and the long-run expected loss per unit total time $l_A(N)$ under policy N, which is given by

$$l_A(N) = \left\{(c_o - w)\left\{\frac{N}{2}\left[2\mu_{X_1} - (N-1)d_a\right]\right\} + c_f\left\{\frac{N-1}{2}\left[2\mu_{Y_1} - (N-2)d_b\right]\right\} + c_{RN}u_{RN}\right\} \Big/ \left\{\frac{N}{2}\left[2\mu_{X_1} - (N-1)d_a\right] + \frac{N-1}{2}\left[2\mu_{Y_1} - (N-2)d_b\right] + u_{RN}\right\}. \tag{49.73}$$

Correspondingly, two GP models obtained by *Lam* [49.14] by putting $d_b = 0$ in (49.63) and $d_a = d_b = 0$ in (49.64) are given by

$$l_G(T) = \frac{(c_o - w)T + c_f\mu_{Y_1}\sum_{j=1}^{\infty}\frac{F_j(T)}{r_b^{(j-1)}} + c_{RT}u_{RT}}{T + \mu_{Y_1}\sum_{j=1}^{\infty}\frac{F_j(T)}{r_b^{(j-1)}} + u_{RT}} \tag{49.74}$$

and

$$l_G(N) = \frac{(c_o - w)\mu_{X_1}\sum_{j=1}^{N}\frac{1}{r_a^{(j-1)}}}{\mu_{X_1}\sum_{j=1}^{N}\frac{1}{r_a^{(j-1)}} + \mu_{Y_1}\sum_{j=1}^{N-1}\frac{1}{r_b^{(j-1)}} + u_{RN}} + \frac{c_f\mu_{Y_1}\sum_{j=1}^{N-1}\frac{1}{r_b^{(j-1)}} + c_{RN}u_{RN}}{\mu_{X_1}\sum_{j=1}^{N}\frac{1}{r_a^{(j-1)}} + \mu_{Y_1}\sum_{j=1}^{N-1}\frac{1}{r_b^{(j-1)}} + u_{RN}}, \tag{49.75}$$

where F_j is the cumulative distribution function of $\sum_{i=1}^{j} X_i$; μ_{X_1} is the mean operating time after installation; μ_{Y_1} is the mean repair time after the first failure; d_a (or r_a) and d_b (or r_b) are the common differences (or ratios) corresponding to the failure and repair processes of a system, respectively; c_o is the average operating cost rate; c_f is the average repair cost rate; c_{RT} (or c_{RN}) is the average replacement cost rate under policy T (or N); u_{RT} (or u_{RN}) is the average replacement downtime under policy T (or N); and w is the average revenue rate of a working system.

Notice that model (49.72) (or (49.74)) only depends on the AP (or GP) through the parameters d_b (or r_b) and μ_{Y_1}, and model (49.73) (or (49.75)) only on d_b (or r_b) and μ_{Y_1} plus d_a (or r_a) and μ_{X_1}. When $K > 1$, μ_{X_1} and μ_{Y_1} are replaced by $\mu_{\bar{X}_1}$ and $\mu_{\bar{Y}_1}$. In practice, model (49.73) or (49.75) is adopted because of its much simpler form. Moreover, under some mild conditions, *Lam* [49.38] has proved that the optimal policy N^* is better than the optimal policy T^*. Note that, under the same conditions, *Zhang* [49.41] has showed that

the optimal bivariate replacement policy $(T, N)^*$ is better than N^*, which in turn is better than T^* (see also *Leung* [49.12]).

The fourth note is that an AP, GP or AGP approach has not incorporated the dependency of data on maintenance actions. If a GP model is appropriate, then the dependency of data upon maintenance actions should be modeled, i. e. the common ratios r_a and r_b of two distinct GPs are two functions of some preventive maintenance (PM) policy, where the subscripts 'a' and 'b' correspond to the failure and repair processes of a system respectively. *Leung* [49.9] established one type of the relationships between the common ratios r_a and r_b and a nonperiodic PM policy.

The fifth note is that a GP model has been widely used in maintenance problems of one-component systems, and two-component series, parallel and standby systems; see *Lam* [49.42], *Lam* and *Zhang* [49.43, 44], and *Zhang* [49.45] for details. *Lam* et al. [49.46] proved that the monotone process model for the multi-state system is equivalent to a GP model for a two-state one-component system by showing that two systems will have the same long-run expected loss per unit total time and the same optimal policy N^*. Furthermore, *Lam* [49.15], *Lam* and *Chan* [49.36], and *Lam* et al. [49.16] also applied the GP to the analysis of data from a series of events. *Lam* [49.47] gave a brief review and more references for the GP. For more properties and applications of GP, see *Lam* et al. [49.16, 48, 49] and *Zhang* et al. [49.50–53]. Finally, the author considers that almost all variants of GP formulation are also valid for AP or AGP.

49.A Appendix

To determine the line of best fit to the N paired-observations, we minimize the sum of squared errors $(S_{\varepsilon\varepsilon})$ given by

$$\begin{aligned} S_{\varepsilon\varepsilon} &= \sum_{k=1}^{K}\sum_{n=1}^{N_k}(y_{n,k}-\hat{y})^2 \\ &= \sum_{k=1}^{K}\sum_{n=1}^{N_k}(y_{n,k}-\hat{\beta}_0-\hat{\beta}_1 x_{n,k})^2 \,. \end{aligned} \tag{49.A1}$$

Denote

$$N = \sum_{k=1}^{K} N_k \,, \tag{49.30}$$

$$\bar{x} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}}{N} \quad \text{and} \quad \bar{y} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} y_{n,k}}{N} \,, \tag{49.A2}$$

$$\begin{aligned} S_{xx} &= \sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}^2 - N\bar{x}^2 \,, \\ S_{yy} &= \sum_{k=1}^{K}\sum_{n=1}^{N_k} y_{n,k}^2 - N\bar{y}^2 \quad \text{and} \\ S_{xy} &= \sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}y_{n,k} - N\bar{x}\bar{y} \,. \end{aligned} \tag{49.A3}$$

Differentiating (49.A1) with respect to $\hat{\beta}_0$ and $\hat{\beta}_1$, setting them equal to zero and solving the associated equations simultaneously, we obtain

$$\hat{\beta}_0 = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} y_{n,k} - \hat{\beta}_1 \sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}}{N} = \bar{y} - \hat{\beta}_1\bar{x} \tag{49.A4}$$

and

$$\hat{\beta}_1 = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}y_{n,k} - \bar{y}\sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}}{\sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}^2 - \bar{x}\sum_{k=1}^{K}\sum_{n=1}^{N_k} x_{n,k}} = \frac{S_{xy}}{S_{xx}} \,. \tag{49.A5}$$

Substituting (49.A4) and (49.A5) into (49.A1), and after some manipulation, we obtain

$$S_{\varepsilon\varepsilon} = S_{yy} - \hat{\beta}_1 S_{xy} \,. \tag{49.A6}$$

It can be shown that

$$\hat{\sigma}_\varepsilon^2 = \frac{S_{\varepsilon\varepsilon}}{N-2} \,, \tag{49.A7}$$

usually called the mean squared error (MSE), provides a good estimator for σ_ε^2, and that

$$t = \frac{\hat{\beta}_1 - \beta_{1,0}}{\hat{\sigma}_\varepsilon/\sqrt{S_{xx}}} \,, \tag{49.A8}$$

a Student's t distribution with $(N-2)$ degrees of freedom. This statistic is used to test a hypothesis that β_1 equals some particular numerical value, say $\beta_{1,0}$.

Now, putting $y = A_{n,k}$ or $\ln G_{n,k}$, $x_{n,k} = n-1$, $\beta_0 = \alpha_A$ or α_G, $\beta_1 = -d$ or $-\ln r$, $\beta_{1,0} = 0$, $\hat{\sigma}_{\varepsilon}^2 = \hat{\sigma}_{A,\varepsilon}^2$ or $\hat{\sigma}_{G,\varepsilon}^2$, $t = t_A$ or t_G, and using (49.A2) through (49.A8), we obtain (49.27) to (49.29) and (49.31) accordingly. The final forms require the following

$$\bar{x} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k}(n-1)}{N} = \frac{\sum_{k=1}^{K}N_k(N_k-1)}{2N} = \frac{\sum_{k=1}^{K}N_k^2 - N}{2N}$$

and

$$S_{xx} = \sum_{k=1}^{K}\sum_{n=1}^{N_k}(n-1)^2 - N\bar{x}^2 = \sum_{k=1}^{K}\frac{(N_k-1)N_k(2N_k-1)}{6} - N\bar{x}^2 .$$

Note that there are three and two estimators for $\mu_{\bar{A}_1}$ and $\sigma_{\bar{A}_1}^2$, respectively, when $d \neq 0$, but five and two estimators for $\mu_{\bar{G}_1}$ and $\sigma_{\bar{G}_1}^2$, respectively, when $r \neq 1$.

References

49.1 K. N. F. Leung, Y. M. Lee: Using geometric processes to study maintenance problems for engines, Int. J. Ind. Eng. **5**, 316–323 (1998)

49.2 K. N. F. Leung, C. Y. Fong: A repair–replacement study for gearboxes using geometric processes, Int. J. Qual. Reliab. Manage. **17**, 285–304 (2000)

49.3 K. N. F. Leung, L. M. A. Cheng: Determining replacement policies for bus engines, Int. J. Qual. Reliab. Manage. **17**, 771–783 (2000)

49.4 K. N. F. Leung: Statistical inference for *K* independent, homogeneous arithmetic processes, Int. J. Reliab. Qual. Safety Eng. **7**, 223–236 (2000)

49.5 K. N. F. Leung: Optimal replacement policies determined using arithmetico-geometric processes, Eng. Optim. **33**, 473–484 (2001)

49.6 K. N. F. Leung, L. F. Kwok: Using arithmetic processes to study maintenance problems for engines, Proc. 2001 Spring National Conf. Operational Research Society of Japan, 156–162 (May, 2001)

49.7 K. N. F. Leung, K. K. Lai, W. K. J. Leung: A comparison of estimators of an arithmetic process using simulation, Int. J. Modeling Simul. **22**, 142–147 (2002)

49.8 K. N. F. Leung: Statistical inference for an arithmetic process, Ind. Eng. Manage. Syst. Int. J. **1**, 87–92 (2002)

49.9 K. N. F. Leung: Optimal replacement policies subject to preventive maintenance determined using geometric processes, Proc. Int. Conf. Maintenance Societies, 1–7 (May, 2002)

49.10 K. N. F. Leung, K. K. Lai: A case study of bus-gearboxes maintenance using arithmetic processes, Ind. Eng. Management Systems International J. **2**, 63–70 (2003)

49.11 K. N. F. Leung: Statistically inferential analogies between arithmetic and geometric processes, Int. J. Reliab. Qual. Safety Eng. **12**, 323–335 (2005)

49.12 K. N. F. Leung: A note on "A bivariate optimal replacement policy for a repairable system", Engineering Optimization (2006) (in press)

49.13 K. N. F. Leung, K. K. Lai: Simulation for evaluating various estimators of *K* independent, homogeneous arithmetic or geometric processes, Int. J. Modeling Simul. (2006) (in press)

49.14 Y. Lam: A note on the optimal replacement problem, Adv. Appl. Probab. **20**, 479–482 (1988)

49.15 Y. Lam: Non-parametric inference for geometric processes, Commun. Statist. **21**, 2083–2105 (1992)

49.16 Y. Lam, L. X. Zhu, J. S. K. Chan, Q. Liu: Analysis of data from a series of events by a geometric process model, Acta Math. Appl. Sin. **20**, 263–282 (2004)

49.17 D. R. Cox, P. A. W. Lewis: *The Statistical Analysis of Series of Events* (Chapman Hall, London 1966)

49.18 H. E. Ascher, H. Feingold: *Repairable Systems Reliability* (Marcel Dekker, New York 1984)

49.19 R. E. Barlow, F. Proschan: *Mathematical Theory of Reliability* (Wiley, New York 1965)

49.20 S. M. Ross: *Applied Probability Models with Optimization Applications* (Holden–Day, San Francisco 1970)

49.21 A. Birolini: *On the Use of Stochastic Processes in Modeling Reliability Problems* (Springer, Berlin Heidelberg New York 1985)

49.22 S. E. Rigdon, A. P. Basu: The power law process: A model for the reliability of repairable systems, J. Qual. Technol. **21**, 251–260 (1989)

49.23 R. E. Barlow, L. C. Hunter: Optimum preventive maintenance policies, Oper. Res. **8**, 90–100 (1960)

49.24 E. J. Muth: An optimal decision rule for repair vs. replacement, IEEE Trans. Reliab. **R-26**, 179–181 (1977)

49.25 K. S. Park: Optimal number of minimal repairs before replacement, IEEE Trans. Reliab. **R-28**, 137–140 (1979)

49.26 T. Nakagawa, M. Kowada: Analysis of a system with minimal repair and its application to replacement policy, Eur. J. Oper. Res. **12**, 176–182 (1983)

49.27 M. Brown, F. Proschan: Imperfect repair, J. Appl. Probab. **20**, 851–859 (1983)

49.28 M. Kijima, H. Morimura, Y. Suzuki: Periodical replacement problem without assuming minimal repair, Eur. J. Oper. Res. **37**, 194–203 (1988)

49.29 M. Kijima: Some results for repairable systems with general repair, J. Appl. Probab. **26**, 89–102 (1989)

49.30 H. Pham, H. Wang: Imperfect maintenance, Eur. J. Oper. Res. **94**, 425–438 (1996)

49.31 H. Z. Wang, H. Pham: Optimal imperfect maintenance models. In: *Handbook of Reliability Engineering*, ed. by H. Pham (Springer, London 2003) pp. 397–416

49.32 D. N. Gujarati: *Basic Econometrics*, 2nd edn. (McGraw–Hill, New York 1988)

49.33 P. P. Ramsey, P. H. Ramsey: Simple tests of normality in small samples, J. Qual. Technol. **22**, 299–309 (1990)

49.34 D. J. Hand, F. Daly, A. D. Lunn, K. J. McConway, E. Ostrowski: *A Handbook of Small Data Sets* (Chapman Hall, London 1994)

49.35 D. F. Andrews, A. M. Herzberg: *Data: A Collection of Problems from Many Fields for the Student and Research Worker* (Springer, New York 1985)

49.36 Y. Lam, S. K. Chan: Statistical inference for geometric processes with lognormal distribution, Comput. Statist. Data Anal. **27**, 99–112 (1998)

49.37 T. Nakagawa: A summary of discrete replacement policies, Eur. J. Oper. Res. **17**, 382–392 (1984)

49.38 Y. Lam: A repair replacement problem, Adv. Appl. Probab. **22**, 494–497 (1990)

49.39 L. M. Leemis: *Probability Models and Statistical Methods* (Prentice-Hall, London 1995)

49.40 P. A. Scarf: On the application of mathematical models in maintenance, Eur. J. Oper. Res. **99**, 493–506 (1997)

49.41 Y. L. Zhang: A bivariate optimal replacement policy for a repairable system, J. Appl. Probab. **31**, 1123–1127 (1994)

49.42 Y. Lam: Calculating the rate of occurrence of failures for continuous-time Markov chains with application to a two-component parallel system, J. Oper. Res. Soc. **46**, 528–536 (1995)

49.43 Y. Lam, Y. L. Zhang: Analysis of a two-component series system with a geometric process model, Naval Res. Logistics **43**, 491–502 (1996)

49.44 Y. Lam, Y. L. Zhang: Analysis of a parallel system with two different units, Acta Math. Appl. Sin. **12**, 408–417 (1996)

49.45 Y. L. Zhang: An optimal geometric process model for a cold standby repairable system, Reliab. Eng. Syst. Safety **63**, 107–110 (1999)

49.46 Y. Lam, Y. L. Zhang, Y. H. Zheng: A geometric process equivalent model for a multi-state degenerative system, Eur. J. Oper. Res. **142**, 21–29 (2002)

49.47 Y. Lam: A geometric process maintenance model, Southeast Asian Bull. Math. **27**, 295–305 (2003)

49.48 Y. Lam, Y. L. Zhang: A geometric-process maintenance model for a deteriorating system under a random environment, IEEE Trans. Reliab. **R-52**, 83–89 (2003)

49.49 Y. Lam, Y. L. Zhang, Q. Liu: A geometric process model for *M/M/*1 queuing system with a repairable service station, Eur. J. Oper. Res. **168**, 100–121 (2006)

49.50 Y. L. Zhang, R. C. M. Yam, M. J. Zuo: Optimal replacement policy for a deteriorating production system with preventive maintenance, Int. J. Syst. Sci. **32**, 1193–1198 (2001)

49.51 Y. L. Zhang: A geometric-process repair model with good-as-new preventive repair, IEEE Trans. Reliab. **R-51**, 223–228 (2002)

49.52 Y. L. Zhang, R. C. M. Yam, M. J. Zuo: Optimal replacement policy for a multi-state repairable system, J. Oper. Res. Soc. **53**, 336–341 (2002)

49.53 Y. L. Zhang: An optimal replacement policy for a three-state repairable system with a monotone process model, IEEE Trans. Reliab. **53**, 452–457 (2004)

50. Six Sigma

The first part of this chapter describes what Six Sigma is, why we need Six Sigma, and how to implement Six Sigma in practice. A typical business structure for Six Sigma implementation is introduced, and potential failure modes of Six Sigma are also discussed. The second part describes the core methodology of Six Sigma, which consists of five phases, i. e., define, measure, analyze, improve, and control (DMAIC). Specific operational steps in each phase are described in sequence. Key tools to support the DMAIC process including both statistical tools and management tools are also presented. The third part highlights a specific Six Sigma technique for product development and service design, design for Six Sigma (DFSS), which is different from DMAIC. DFSS also has five phases: define, measure, analyze, design and verify (DMADV), spread over product development. Each phase is described and the corresponding key tools to support each phase are presented.

In the forth part, a real case study on printed circuit board (PCB) improvement is used to demonstrate the application of Six Sigma. The company and process background is provided. The DMAIC approach is specifically followed and key supporting tools are illustrated accordingly. At the end, the financial benefit of this case is realized through the reduction of cost of poor quality (COPQ).

Finally, last part is given over to a discussion of future prospects and conclusions.

Since the early 1990s, Six Sigma swept the business world, driving an unprecedented emphasis on greater manufacturing and service quality. Six Sigma is one of the few quality initiatives that actually originated from industrial practice. Six Sigma was originally devised as a measure of quality that strives for near perfection. It has developed into a disciplined, data-driven, customer-focused approach to reduce defects and bring about substantial financial growth. Although most Six Sigma efforts were focused on manufacturing operations in the early years, the Six Sigma approach has now been more widely used in non-manufacturing industrial sectors such as finance, insurance, health care, and telecommunications. Users include American Express, American International Group (AIG), Bank of America, Citibank, J.P. Morgan, Chase, Merrill Lynch, Vanguard, etc. These companies have actually seen larger business impacts and cost savings than those in manufacturing.

50.0.1 What is Six Sigma?

Motorola first introduced the Six Sigma program in the late 1980s with the aim of increasing profitability by reducing defects. General Electric (GE) followed the approach at their manufacturing sites and later at their financial service divisions. After that, Six Sigma was thought to be applicable to all processes and transactions within GE. Six Sigma has now evolved from a quality

Table 50.1 Final yield for different sigma levels in multistage processes

Average sigma level	1	2	3	4	5	6
Final yield for 10 stages	0.0%	2.5%	50.1%	94.0%	99.8%	100.0%
Final yield for 100 stages	0.0%	0.0%	0.1%	53.6%	97.7%	100.0%
Final yield for 1000 stages	0.0%	0.0%	0.0%	0.2%	79.2%	99.7%

improvement program to an overall business strategy executive system and business-results-oriented program, which seems more *total* than total quality management (TQM). We will describe the basic definition of Six Sigma in this section and will elaborate its systematic methodology and business structure in later sections.

Six Sigma is both a business improvement strategy and a methodology to measure process performance. It is used to increase profits by eliminating defects, waste, and variability and to find the causes of mistakes in products, processes and services to increase yields. In Six Sigma, focus on the customer is the top priority. Performance standards are based on actual customer input, so that process effectiveness can be measured and customer satisfaction can be predicted.

In terms of business process improvement, variation reduction is the key since variation signals fluctuation in the process output and is often a major source of poor quality. Variation is present in all processes and every aspect of work. Unintended variation reduces process performance and decreases customer satisfaction. Because of the existence of variation, producing high-quality products and services in the modern industrial environment is a tough task.

Therefore, Six Sigma aims particularly at reducing variation. The word *sigma* or the symbol "σ" is used in statistical notation to represent the standard deviation in a population. The standard deviation is also used as a general measure of variation in any kind of product or process. With six standard deviations between the process mean and the customer's specification limit, we arrive at 3.4 defects per million opportunities (DPMO); that is, a 99.9997 percent yield. Before the Six Sigma technique was introduced, a three-sigma level of variation was regarded as being fairly good quality performance. Three sigma may be acceptable for a product or process having only a single or a few stages. It is not good enough for many products that are the result of hundreds of thousands of stages, such as automobiles and computers.

For example, if a production process is made up of ten stages where the yield of each stage is as high as 90%, the probability of producing a satisfactory product in the first run would be $0.9^{10} = 35\%$. This indicates that about 65% of the products are defective. If a production process is made up of 100 stages, the probability of producing a satisfactory product under the three-sigma program could be as low as 0.1%, as shown in Table 50.1. The Six Sigma regime, however, allows only 3.4 defects for every million opportunities, which ensures a quality product even if the process involves a large number of stages (Table 50.1). Part of the reason for using such a strict requirement in quality management is actually to accommodate the common multistage processes in modern industrial practice.

50.0.2 Why Six Sigma?

The successful implementation of Six Sigma can result in benefits in the areas of cost reduction, increased profit, increased market share and enhanced business competitiveness, mainly by the reduction of the cost of poor quality (COPQ).

COPQ usually includes appraisal costs, internal failure costs, and external failure costs. Appraisal and inspection costs are often incurred, for example, in checking finished goods before they leave the factory, inspecting purchased equipment/supplies, proofreading financial and legal documents, reviewing charges prior to billing, etc. Internal failure costs are those for repairing, replacing, or discarding work in progress or completed work before the delivery of the product to the customer. External failure costs are those that directly affect the customer and are the most expensive to correct, including tangible remedial costs and the intangible costs associated with losing dissatisfied customers.

COPQ cannot be underestimated. In manufacturing industries, COPQ sometimes reaches 15% of total sales (source: Six Sigma Academy). In service industries, the situation is even more serious. COPQ may account for as much as 50% of total costs.

However, these COPQ could be saved with the use of Six Sigma. General Electric has estimated savings of 2 billion US dollars during the first five years of Six Sigma implementation, and Allied Signal has estimated savings of 1.1 billion US dollars in two years. Indeed, thousands of companies around the world have enjoyed the breakthrough benefits of Six Sigma.

For example, Legend Computers in China reported in 2002 savings of $20 million dollars during the first year of implementation. In the same year, the International Bank of Asia in Hong Kong reported savings of 1.4% of total costs during the first year of Six Sigma implementation.

50.0.3 Six Sigma Implementation

Six Sigma implementation is usually a top-down approach, i. e., from the strong commitment of top management. As most Six Sigma projects span several departments, organizational barriers could not be removed without leadership commitment to Six Sigma. Strong commitment, leadership and strategic involvement have proven to be key factors for Six Sigma's success. Secondly, as Six Sigma requires a long-term mentality, it needs to be positioned first as a strategic initiative and then be linked to operational goal. It is important to tie the Six Sigma implementation to corporate goals, such as increased profits through lower costs and higher loyalty, for example. Also, effective internal communication is another key issue for the success of Six Sigma implementation.

In the following, a typical business structure for Six Sigma implementation is introduced. Several potential failure modes and practical considerations of Six Sigma implementation are also discussed.

Training and Belt Structure

The deployment of Six Sigma in a company usually starts with education. Without the necessary training, people are not able to bring about Six Sigma breakthrough improvements. Six Sigma establishes well-defined and structural roles and responsibilities for a project team, and team members are given formal training according to their roles to help the team work effectively. A Six Sigma team is usually organized in a belt structure (as in martial arts) as follows.

At the top of the belt structure is the Six Sigma executive. The Six Sigma executive could be a council that consists of top managers who have the vision and make strategic decisions for a company. They are responsible for establishing the roles and structures of Six Sigma projects. They also need to make decisions on project selection and resources allocations. A progress review is conducted periodically to monitor projects.

Champions are the senior managers who supervise Six Sigma projects. They report directly to the Six Sigma executive and represent the team to the executive. They also need to seek resources and to learn the focus of the business from the Executive. In addition, champions meet black belts and green belts periodically to review the progress and coach the team.

Master black belts work with the champions to ensure that Six Sigma objectives and targets are set. Meanwhile, they are the statistical problem-solving experts in a company. Their responsibilities include determining plans, providing technical expertise, training and coaching black and green belts.

Black belts, as on-site Six Sigma experts, usually possess the technical background needed to help green belts and the other team members to understand the project and apply appropriate statistical techniques. Their roles are to provide formal training to local personnel in new strategies and tools, provide one-on-one support to local personnel, pass on new strategies and tools in the form of training, workshops, case studies, local symposia, etc., and find application opportunities for breakthrough strategies and tools, both internal and external (i. e., to the suppliers and customers).

Green belts, on the other hand, execute Six Sigma in their specific area as a part of their overall job. They may assist black belts in completing sections of their projects and apply their learning to their daily performance of their jobs.

According to the Six Sigma Academy, black belts are able to save companies approximately US$230 000 per project and can complete four to six projects per year. The American Society for Quality (ASQ) has been certifying Six Sigma black belts (SSBB) internationally in recent years. Up to the middle of 2002 there were around 200 ASQ-certified black belts in the US and only 11 ASQ-certified black belts outside the US. Among them, there was only one in the greater China area (Table 50.2).

Six Sigma Failures (Sick Sigma)

Although Six Sigma is a powerful approach, it can lead to failure when some critical issues are neglected. How-

Table 50.2 Number of Six Sigma black belts certified by the American Society for Quality (ASQ) internationally (ASQ record up to April, 2002)

Indonesia	1	United Kingdom	1
India	5	Hong Kong	1
Japan	1	Mainland China	0
Australia	1	Taiwan	0
Brazil	1	Singapore	0

ever, as more companies have implemented Six Sigma since the 1990s, the factors that have led to failure have been identified and summarized. According to *Snee* and *Hoerl* [50.1], project selection and management support are usually the two main sources of failure.

The failure modes in project selection usually include projects not tied to financial results, poorly defined project scopes, metrics, and goals, projects lasting more than six months, the wrong people assigned to projects, project teams that are too large, and infrequent team meetings. On the other hand, the failure modes in management support may include black belts with little time to work on projects, poor or infrequent management reviews, poor support from finance, information technology (IT), human resource (HR), etc., and poor communication of initiatives and progress [50.1].

Especially, for a Six Sigma program to sustain without failure, recognition and reward systems are the key. If recognition and reward systems are lacking or remain unchanged, the program cannot last. Necessary practices include establishing and using selection and promotion criteria and developing corresponding performance management and reward systems. GE's approach, which links 40% of management bonus to Six Sigma, may be too aggressive, but a company must adequately compensate those high-performing members.

Note that the use of statistical methods is not on the list of major failure modes. With recent advances in information technology, computing and sensing technology, the use of advanced statistical methods has become handy via commercial software packages (such as MINITAB, JMP, etc.). Therefore, the use of statistical tools is no longer a bottleneck in Six Sigma implementation.

Moreover, various industry types and company natures are also not an excuse for Six Sigma failure. Six Sigma has been successfully applied to many processes outside of manufacturing, regardless of the company size or nature of the industry. In particular, transactional processes, such as software coding, billing, customer support, etc., often contain variation or excessive cycle time and can be optimized by applying Six Sigma. For example, HR managers may apply it to reduce the cycle time for hiring employees, and regional sales may apply it to improve forecast reliability, pricing strategies or variations.

50.1 The DMAIC Methodology

50.1.1 Introduction

The development of Six Sigma is evolutionary, not revolutionary, and it integrates many useful quality management tools. Thus, it is not surprising to find overlaps between the Six Sigma, TQM, lean, and ISO approaches. The core methodology of Six Sigma is driven by close understanding of customers' needs and the disciplined use of facts, data and statistical analysis, which consists of five phases, i. e., define, measure, analyze, improve, and control (DMAIC).

In the define phase, the specific problem is identified, and the project goals and deliverables are defined. In the measure phase, the critical-to-quality (CTQ) characteristics are identified and the measurement system is reviewed. The nature and properties of the data collection have to be understood thoroughly to ensure the quality of the data. In the analyze phase, both quantitative (i. e., statistical) methods and qualitative (i. e., management) tools are used to isolate the key information that is important to explaining defects. In the improve phase, the key factors that cause the problem should be discovered. In the control phase, the key factors and processes are controlled and monitored continuously to ensure that the improvement is sustainable and the problem will not occur again. A detailed case study on the implementation of the DMAIC methodology in printed circuit board manufacturing can be found in *Tong* et al. [50.2]. The paper "Six Sigma approach to reducing fall hazards among cargo handlers working on top of cargo containers: a case study" by *Ng* et al. [50.3, 4] is another case study using DMAIC that focuses on a non-manufacturing case.

50.1.2 The DMAIC Process

More specifically, we implement the DMAIC methodology in detailed steps in sequence to shift our focus from the output performance (i. e., y) to the root causes (i. e., the x). Based on these steps, we transfer a practical problem into a statistical problem (e.g., mapping x and y), find out a statistical solution for that [e.g., solving $y = f(x)$] and then transform the statistical solution into a practical solution. Each step is described in the following, and the corresponding key tools will be further explained in a later section.

Phase 1: Define (D)

This phase defines the Six Sigma project, which includes a problem statement, the objectives of the project, the expected benefits, the team structure and the project time line. At the end of this phase, we should have a clear operational definition of the project metrics for the final evaluation. In this phase, the main tasks are to identify who the customer is, select the project area, define the goal, scope and resources of the project, form a Six Sigma project team, define the team members' responsibilities, and estimate the profit and cost for this project to ensure the value of the project. Key tools in this phase include the project charter; business process mapping; suppliers, inputs, process, outputs and customer (SIPOC); etc.

Phase 2: Measure (M)

By taking steps in the measure phase, we have a clear understanding of the performance of the current process and, only after knowing where we are now, can we determine where we should be in the future. Three implementation steps in this phase are to select the critical-to-quality (CTQ) measures, determine deliverables, and quantify the measurability of y.

Select the Critical to Quality (CTQ) Measures. In this step, we will identify the external CTQ from the customer's point of view (i. e., the big Y) that will be improved, and then link that with the internal CTQ (i. e., the small y), which is a quantitative measure in the company and will be the focus of the project. Key tools in this step include customer needs mapping (CNM), quality function deployment (QFD), failure modes and effects analysis (FMEA), etc.

Deliverables. We will establish a performance standard and develop a data collection plan for the internal CTQ y in this step. If the measure of y from the previous step is attributal, what is the definition of a defect? If the data are continuous, what are the lower and upper specifications for defectiveness? Key tools used in this step include process mapping and yield calculation.

Quantify Measurability. We validate the measurement system on y to ensure the measurement results are accurate for the following analysis. We may need to improve the measurement system before continuing. Key tools include measurement system analysis (MSA), gage repeatability and reproducibility (R&R) study.

Phase 3: Analyze (A)

After we identify the y in the process, we need to determine the x (root causes), which may impact on the performance of the y. In the analyze phase, we use various management and statistical tools to discover the x for future improvements. Three implementation steps in this phase are to establish the baseline, determine the improvement plan, and identify the sources of variation.

Establish the Baseline. We will establish the process capability for the current process to understand where we are now. We need to collect the current process data, use graphical tools to analyze the data, and calculate the process capability indices, the defect per million opportunities (DPMO), and the sigma level (Z). Key tools include: histograms, process capability indices (PCI), etc.

Determine Improvement Plan. We quantify the goals for the improvement to make the aim of the project clear, and we may determine if the goal is significantly different from today's performance (i. e., the baseline) through hypothesis testing. Key tools include benchmarking, hypothesis testing, t-test, analysis of variations (ANOVA), etc.

Identify Variation Sources. We list all the potential factors (x) that may influence the performance of y. Regression analysis may be conducted, where applicable, to identify potential x. Key tools include brainstorming, cause-and-effect diagram, regression analysis, etc.

Phase 4: Improve (I)

As the root causes for variation are obtained, it becomes possible for us to fix these root causes. In the improve phase, the way that we can achieve a better process needs to be found, where the design of experiments (DOE) is a key technique to help us quantify the relation between the ys and xs, and to improve the process by finding the optimal setting of xs for each y. In this phase, we follow three implementation steps: screen potential sources of variation, discover variable relationships, and formulate the implementation plan.

Screen Potential Sources of Variation. We determine the few vital xs from the many trivial xs in this step. DOE is a key tool for factor screening. Both full factorial and fractional factorial experiments can be used. If necessary, historical data can be used with care, and a similar model or simulation may be used as well.

Discover Variable Relationships. We develop the transfer function $[y = f(x)]$ linking the y to the vital xs. Based on this, we then determine and confirm the optimal settings for the vital few xs. DOE is a key tool for characterization and optimization as well. Various DOE techniques, such as the response surface method (RSM), robust design and the Taguchi method, can be applied in this step. Other than that, simulation or surveys can also be used to find the relationship.

Formulate Implementation Plan. In this step, if a new process or process steps have been put in place, show the new process map. For the new process, indicate the new in-process measurements and associated specifications. If there is not a new process, indicate any new measurements put in place. We list how the changes to the xs will be implemented and how much flexibility is available in the settings of each x. Key tools in this step include tolerance design, main effects plots, interaction plots.

Phase 5: Control (C)

After determining how to fix the process, we want the improvement for the process to be sustainable. The control phase is set up to ensure sustainable improvement and to deploy measurement tools to confirm that the process is in control. It is also critical to realize the financial benefits and develop a transfer plan in this phase. Three implementation steps include validating the implementation plan, controlling the inputs and monitoring the outputs, and finally sustaining the change.

Validate the Implementation Plan. To determine how well the xs can be controlled, we will validate the measurement system on the xs, and we may need to improve measurement system before continuing. We will also report new sigma levels and new DPMO levels at this step. Key tools include gage R&R, ANOVA, etc.

Control Inputs and Monitor Outputs. We determine how each vital xs can be controlled (e.g., attribute control chart, variable control chart, mistake-proofing, etc.) and set up a monitoring plan for the y and xs in this step. Key tools include statistical process control (SPC), attribute control charts, variable control charts, Poka–Yoke (mistake-proofing), etc.

Sustain the Change. The objective of this step is to ensure that changes last after the improvement strategy has been implemented. Process control plans need to be developed and implemented for each x. We will also verify the financial gains that can be achieved and if this project is translatable to any other regions, lines, sites, processes, etc. Key tools in the final step include out-of-control plans, mistake-proofing, audit strategy, etc.

50.1.3 Key Tools to Support the DMAIC Process

This section presents the key tools to support the DMAIC process. Only a few key tools can be covered in this section and each method is outlined briefly with the basic ideas and mechanisms. The books and papers cited in this section give more details.

Business Process Mapping (SIPOC Diagrams)

Purpose. SIPOC stands for suppliers, inputs, process, outputs and customer. SIPOC diagrams are graphical tools to identify all relevant elements of a business process and map the process flow before the project begins. They are usually used in the define phase.

Definitions.

Supplier: Whoever produces, provides, or furnishes the products or services for the input of the process, either an internal or an external supplier.

Inputs: Material, resources and data required to execute the process.

Process: A collection of activities that take one or more kinds of input and creates output that is of value to the customer.

Outputs: The tangible products or services that result from the process.

Customer: Whoever receives the outputs of the process, either an internal customer or an external customer.

How to do it.

Step 1. Clear statement of CTQ and the process.
Step 2. Clear statement of start/end point.
Step 3. Identify major customers, suppliers, outputs, and inputs.
Step 4. Identify the five to seven major process steps using brainstorming and storyboarding.
Step 5. Decide what step to map in detail.
Step 6. Complete detailed map.

Quality Function Deployment (QFD)

QFD is a systematic approach to prioritize and translate customer requirements (i. e., external CTQ) into appro-

priate company requirements (i. e., internal CTQ) at each stage from product development to operations to sales and marketing to distribution. This method is usually used in the measure phase. It is also useful in design for Six Sigma (DFSS) and will be introduced in more detail in the DFSS section.

Failure Modes and Effects Analysis (FMEA)

Purpose. FMEA is a tool to reduce the risk of failures. It is also a tool to identify and prioritize CTQ at the measure phase.

Definitions.

Severity: the assessment of how severe a failure mode is. The severity usually scales from 1–10. Scale 1 means a minor failure mode that may not be noticed, and 10 means a very serious failure that may affect safe operations.

Occurrence: The likelihood that a specific cause will result in the failure mode, which scales from 1–10 with 10 being the highest likelihood.

Detection: The assessment of the ability to identify the failure mode. A 1–10 scale is often used with 10 being the lowest detectability.

RPN: The risk priority number (RPN) is the output of a FMEA. RPN = Severity × Occurrence × Detection.

How to do it [50.4].

Step 1: Identify the products, services, or processes.
Step 2: Identify the potential failure that would arise in the target process.
Step 3: Identify the causes of the effects and their likelihood of occurrence.
Step 4: Identify the current controls for detecting each failure mode and the ability of the organization to detect each failure mode.
Step 5: Calculate the RPN by multiplying the values of severity, potential causes, and detection.
Step 6: Identify the action for reducing or eliminating the RPN for each failure mode.

Measurement System Analysis (MSA)

Purpose. A statistical evaluation of the measurement system must be undertaken to ensure effective analysis of any subsequent data generated for a given process/product characteristic. MSA is usually used in the measure and control phases to validate the measurement system for the y and xs.

Definitions.

Gage R&R: is a tool to study the variation in the measurement arising from the measurement device and the people taking the measurement.

Repeatability: The variability that reflects the basic, inherent precision of the gage itself.

Reproducibility: The variability due to different operators using the gage (or different time, different environments) [50.5].

How to do it.

Step 1: Collect the data. Generally two to three operators, 10 units to measure, and each unit is measured 2–3 times by each operator.
Step 2: Perform the calculations to obtain %R&R [50.5].
Step 3: Analyze the results. A rule of thumb is that:

- %R&R < 10%: measurement system is acceptable.
- %R&R between 10–30%: measurement system may be acceptable. We will make decisions based on the classification of the characteristics, hard applications, customer inputs, and the sigma level of the process.
- %R&R > 30%: measurement system is not acceptable. We should improve the measurement system by finding problems and removing root causes.

Process Capability Analysis

Purpose. Process capability analysis is a statistical technique to quantify process variability, analyze this variability relative to customer requirements or specifications, and assist in reducing the variability [50.5]. It is used in the analyze phase.

Definitions.

Cp: Process/product capability index, is the relationship of the process/product variation to the upper and lower specification limits. It is related to the potential process capability and not a measure of how centered the data are.

Cpk: It compares process variability with the specification's width and location. It takes into account that the sample mean may be shifted from the target. Since both the mean shift and the variability of the characteristics are considered, Cpk

is better related to the capability of the current process.

How to do it. The detailed calculation and analysis is given by *Montgomery* [50.5].

Cause–Effect Diagram (Fishbone Diagram)

Purpose. This is a graphical brainstorming tool to explore the potential causes (i. e., xs) that result in a significant effect on y. It is usually used in the analyze phase.

How to do it.

Step 1: Define clearly the effect or analyzed symptom (y) for which the possible causes (xs) must be identified.
Step 2: Place the effect or symptom (y) being explained on the right of a sheet of paper.
Step 3: Use brainstorming or a rational step-by-step approach to identify the possible causes.
Step 4: Each of the major areas of possible causes should be connected with the central spine by a line.
Step 5: Add possible causes (xs) for each main area.
Step 6: Check for completeness.

Design of Experiments (DOE)

Purpose. DOE is a major tool in the improve phase. It is used for screening the few, vital xs, characterizing the relationship between y and xs, and optimizing the setting of the vital xs.

Definitions.

Factor: An independent variable (i. e., xs) whose state can be varied.
Level of a factor: The state of the factor.
Full factorial experiments: Discover the factor effects and relationship between y and xs by running all the combinations of factor levels.
Fractional factorial experiments: An economical approach to discovering the factor effects and to screening the vital few xs by running only part of the combinations of factor levels.
Response surface methodology (RSM): A DOE technique that is useful for modeling and optimization in which a response of interest y is influenced by several factors xs and the objective is to optimize this response. This method will be discussed more in the DFSS section.

How to do it [50.6, 7].

Step 1: State the problem.
Step 2: Choose the response variable (y).
Step 3: Choose the factors (xs) and their levels and ranges.
Step 4: Determine the experimental plan (i. e., the design matrix).

1. To screen the xs to obtain the few, vital xs, we often use factorial experiments. In such cases, if the number of runs is moderate and we have enough time and resources, we may conduct a full factorial experiment; if the number of runs is large or time and resources are limited, we may consider a fractional factorial experiment.
2. To obtain the optimal response, we may conduct RSM, which is usually conducted after variable screening.

Step 5: Run the experiments under the prescribed conditions and collect the response data.
Step 6: Analyze the data collected using main effect plots, interaction plots, ANOVA, etc.
Step 7: Conclude the experiment and make recommendations. A confirmation run or a follow-up DOE is usually needed.

Statistical Process Control (SPC)

Purpose. SPC is a major tool in the control phase. It is used to control and monitor the stability and capability of the few, vital xs for CTQ.

How to do it. This method will be discussed in more detail in the DFSS section. For a general introduction to SPC, see *Montgomery* [50.5]. For recent advances in SPC, the reader may refer to http://qlab.ielm.ust.hk and references therein.

50.2 Design for Six Sigma

The success of Six Sigma's DMAIC methodology has generated enormous interest in the business world. One of the basic ideas is to measure existing defective processes quantitatively and then to improve them. Compared with this defect-correction methodology, design for Six Sigma (DFSS) is a proactive methodology, which focuses on the new product/service development to prevent quality defects from appearing instead of solving problems when they happen in existing processes.

DFSS is a disciplined and statistical approach to product and service design that ensures that new designs can meet customer requirements at launch. The objective of DFSS is to eliminate and reduce the design vulnerabilities in both the conceptual and operational phases by employing scientific tools and statistical methods.

Unlike the DMAIC methodology, the phases of DFSS are not universally defined. There are many methodologies, such as Woodford's identify, design, optimize, validate (IDOV), El-haik's identify, characterize, optimize, verify (ICOV), Tennant's define, customer concept, design, and implement (DCCDI), and so on. All these approaches share common themes, objectives, and tools. In this section, we refer to above methodologies, especially General Electric's DFSS approach called define, measure, analyze, design and verify (DMADV):

Define the project goals and customer requirements.
Measure and determine customer needs and specifications.
Analyze the options of conceptual solutions to meet customer needs.
Design the product/service to meet customer needs.
Verify the design performance and ability to meet customer needs.

50.2.1 Why DFSS?

Proactive versus Retroactive

During the product/service design process, conceiving, evaluating and selecting good design solutions are difficult tasks with enormous consequences. Usually organizations operate in two modes: *proactive*, that is, conceiving feasible and healthy conceptual solutions the first time; and *retroactive*, that is, an after-the-fact practice that drives design in a design–test–fix–retest cycle and creates what is broadly known as the *fire-fighting* mode of design. If a company follows this practice, it suffers from high development costs, longer times to market, lower quality levels, and marginal competitive edge [50.8].

Compared to retroactive approaches such as DMAIC, which apply performance improvement in the later stages of the product/service life cycle, DFSS shifts the attention to improving performance in the front-end design stages. That is, the focus is on problem prevention instead of problem solving. This action is motivated by the fact that the design decisions made during the early stages of the product/service life cycle have the largest impact on both total cost and quality of the system. It is often claimed that up to 80% of the total cost is committed in the concept development stage. Also, at least 80% of the design quality is committed in the early design stages. According to a study of the design community [50.8], at the early design stage, the impact (influence) of design activity is much higher than a later stage, while the correction cost in the early stage is much lower.

Experience Dependency versus Scientific and Systematic Methodology

Currently, most design methods are empirical in nature, while the work of the design community is often based on experience. This experience-based tradition often leads to unnecessary variation and is difficult for project manager to control. As a result, vulnerabilities are introduced into the new design that makes it impossible for the product/service to achieve Six Sigma performance. This is another motivation for devising DFSS as a scientific and systematic design method to address such needs.

50.2.2 Design for Six Sigma: The DMADV Process

Generally speaking, DFSS has five phases spread over product development. They are called: define, measure, analyze, design and verify (DMADV).

Phase 1: Define (D)

The process of product/service design begins when there is a need (internal or external), which can be a problem to be solved or a new invention. In this phase, design objectives, scope and available resources should be simply and clearly defined in the design project charter as the key deliverables.

Phase 2: Measure (M)

In particular, the voice of customer (VOC) is the critical input in customer-oriented design. Based on VOC, the internal CTQ measures (critical to quality or critical to satisfaction, i. e., the y), such as cost, performance, reliability, aesthetics and serviceability, need to be identified quantitatively and to be prioritized according to their importance to customers. This kind of information can help to define the function requirements in a later phase.

Phase 3: Analyze (A)

In this phase, the CTQs will be decomposed into measurable and solution-free functional requirements (FRs). Then, a number of conceptual-level design alternatives should be produced by the design team for the FRs, considering cost, physical properties, the difficulties to operate/manufacture and maintenance, etc. Through summarizing the design requirements and conceptual-level design alternatives, an overall set that contains high-potential and feasible solutions can be produced to help the design team to decide on the best solution considering the original design charter including the performance, the constraint of cost and available resources.

Phase 4: Design (D)

Once the design team fixes the selection of the conceptual solutions, they need to decompose the FRs into design parameters (DPs). At the same time, they need to consider the potential risk to achieve CTQs when they create detailed designs to the level of design parameters. Then, optimization tools will be used to get optimal values for the design parameters. In DFSS, optimization can be reached statistically, and by using statistical tools, the transfer functions can be generated to mathematically represent the relationships between the input and output of a product or a service process. Then, the design team can rely on the transfer function to optimize the design solutions so that the product/service can achieve a target performance and be insensitive to uncontrollable factors (noise factors), such as the environment and production case-to-case variation.

Phase 5: Verify (V)

In this phase, the design team makes a model formed by the simulation of a service process or a physical prototype that is the first working version of the product. Based on these few prototypes, the design team evaluates and tests the whole design to predict if the future product's performance can meet the design charter and how to improve the solution when failure occurs.

50.2.3 Key Tools to Support the DMADV Process

Below is a summary of the key tools used to support the DMADV process.

Voice of Customer (VOC)

Purpose. Define customer needs/requirements for the new product/service design or existing product/service redesign.

Input. Market segment defined – who the customers are and their environment.

Output. Detailed customer requirements.

How to do it [50.9].

Step 1: Define market segments – to understand who the customers are and where to go to gather their needs.
Step 2: Identify objective for interviews of customer – to learn what of their needs are new, unique, and difficult (NUD).
Step 3: Select customer groups within the target market segments.
Step 4: Decide on the customer visit team – divide into three roles: leader, subordinate interviewer that helps adding balance and diversity in the discussion, and statement writer that writes down the VOC needs statement.

1. Create an interview guide based on objectives – to get customers' responses that are rich in description of needs.
2. Listen, probe, and observe customers by asking stimulating questions and open-ended statements to gather the VOC. Image data can be gathered by actual observation of customers' responses to existing products or services.

Kawakita Jiro (KJ) Method [50.10]

Purpose. Structure and rank the customer requirements.

Input. The detailed VOC.

Output. Organized customer requirements.

How to do it [50.11].

Step 1: Write down customer descriptions as *statements of customer requirements* on a POST-IT note and put them on the wall.
Step 2: Group the similar customer requirements together.
Step 3: Review the customer requirements statements and throw out redundant ones.
Step 4: Write a summary to express the general idea for each group. For those that do not relate to anything else, label it as *independent*.
Step 5: Vote for the most important groups and rank the top three groups and assign some relationships. If a group supports another group in a positive manner, we add an arrow pointing from the supporting group to the supported group. If the relationship is contradictory, we add a line pointing between the two groups with blocks on the end.
Step 6: Look at each detailed customer requirement and highlight the new, unique, or difficult ones.
Step 7: Ask customers to rank (on a scale of 1–10) the strength of importance for each requirement.

The result of these ranked and structured customer requirements will flow into the QFD process.

Quality Function Deployment (QFD): the houses of quality [50.12]

QFD is a methodology that establishes *bridges* between qualitative, high-level customer needs/requirements and the quantitative engineering terms that are critical to fulfilling these high-level needs. By following QFD, relationships can be explored among customer requirements, CTQ measures, function requirements (FRs), design parameters (DPs) and process variables (PVs). And the priorities of each CTQ, FR, DP and PV can be quantitatively calculated.

Generally, the QFD methodology is deployed through a four-phase sequence.

Phase 1 – critical-to-satisfaction planning (HOQ1)
Phase 2 – functional requirements planning (HOQ2)
Phase 3 – design parameters planning (HOQ3)
Phase 4 – process variable planning (HOQ4)

In this chapter, HOQ1 will be introduced in detail as an example.

Input. Structured and ranked new, unique and difficult (NUD) VOC from the KJ diagram.

Key Output. The priorities of each CTQ.

How to do it [50.8].

Step 1: Convert NUD VOC ("WHATs") into a list of CTQs ("HOWs") in terms of the engineering perspective to support customer requirements along the roof of the house. There may be more then one CTQ to achieve each customer requirement.
Step 2: Quantify the relationship between each customer requirement to each CTQ on a $1-3-9$ scale ($9 =$ strong fulfillment, $3 =$ moderate fulfillment, $1 =$ weak fulfillment, or $0 =$ no relationship). These values help to identify which CTQs are critical and which are not.
Step 3: Identify the correlation between each pair of CTQ to address the cooperative and conflicting relationships among CTQs to develop the design to be as cooperative as possible.
Step 4: Conduct a competitive assessment with a main competitor. The comparison with the key competitor on each customer requirement is on a 1–5 scale, with five being high.
Step 5: Prioritize customer requirements. These priorities include importance to customer from the KJ method, improvement factor, and absolute weight. Customer requirements with low completive assessments and high importance are candidates for improvement, which will be assigned improvement factors on a 1–5 scale, with five being the most essential target to improve. The absolute weight can then be calculated by multiplying the customer importance and the improvement factor.
Step 6: Priority CTQs. The CTQs are prioritized by determining absolute weight and relative weight. The absolute weight is calculated by the sum of the products of the relationship between customer requirements and CTQs and the importance to the customer. The relative weight is the sum of the products of the relationship between customer requirements and CTQs and customer requirement absolute weights. The relative and absolute weights are evaluated to prioritize and select CTQs for improvement.

Furthermore, the design team can apply the same method for identifying the relationship among CTQs, functional requirements, design parameters and process variables.

The Pugh Concept Evaluation and Selection Process [50.13]

The Pugh concept evaluation is a solution-iterative selection process. The method alternates between generation and selection activities. The *generation* activity can be enriched by the TRIZ (theory of inventive problem solving, [50.14]) methodology to generate conceptual solutions for each functional requirement. The *selection* activity can use a scoring matrix called the *Pugh matrix* or the *criteria-based matrix* to evaluate the concepts.

Input. Functional requirements and conceptual solutions to achieve corresponding FRs.

Output. The conceptual solutions, which are selected and ready to go forward into the design phase.

How to do it [50.15].

Step 1: Define concept selection criteria from a clear and complete set of requirements.
Step 2: Define a best-in-class benchmarked datum concept.
Step 3: Provide candidate concepts to evaluate against the datum.
Step 4: Evaluate each concept against the datum using (+)s, (−)s, and (*S*)s to rank the fulfillment of the concept selection criteria.

(+) means the concept is better than the benchmarked datum concept;
(−) means the concept is worse than the benchmarked datum concept;
(*S*) means the concept is the same with the benchmarked datum concept.

Step 5: Refine criteria as necessary during the first cycle of evaluation.
Step 6: Analyze the results from the first cycle of evaluation: the sum of (+)s, (−)s, and (*S*)s.
Step 7: Identify weakness in concepts that can be turned into (+)s.
Step 8: Create hybrid *super-concepts* by integrating the strengths of similar concepts to remove (−)s and (*S*)s.
Step 9: Select a new datum based on the scoring that suggests a superior concept after the first cycle of evaluation.
Step 10: Add any new concepts that have been developed.
Step 11: Repeat the evaluation process through the second cycle.
Step 12: The superior concept is selected and ready to go forward into the development or design phase.

Design Failure Modes and Effects Analysis [50.16]

DFMEA is applied to define qualitatively and rank quantitatively the failure modes and effects for new products and service processes across all the phases of DMADV. In particular, the design team can use DFMEA in a design concept for potential failure modes so it can address them early in the design. Usually DFMEA is conducted on the superior concept, which is chosen from all the candidate concepts in the Pugh concept-selection process.

Input. Superior concept architectures, functional requirements, the physical form, etc.

Output. Causes of failure and corrective action.

How to do it [50.8].

Step 1: Develop a block diagram of the design element or function being analyzed (at system, subsystem, subassembly, or component level).
Step 2: Determine the ways in which each design element or function can fail (failure modes).
Step 3: Determine the effects of the failure on the customer(s) for each failure mode associated with the element or function.
Step 4: Identify potential causes of each failure mode.
Step 5: List the current design controls for each cause or failure mode.
Step 6: Assign severity, occurrence, and detection ratings to each cause.
Step 7: Calculate risk priority numbers (RPN) for each cause.
Step 8: Apply tools and best practices to reduce the sensitivity to root causes of failure or eliminate root causes of failure and recalculate RPNs.
Step 9: Document causes of failure and corrective action results qualitatively and quantitatively.

Response Surface methods [50.6,17]

Purpose. Optimize the system performance in the *design* phase by constructing a statistical model and response surface map that represents the relationship between the response and the critical design parameters. If the design parameters are quantitative and there are only a few of them, RSM is an effective tool for modeling and optimization.

Input. Critical design parameters.

Output. Surface map and equations that determine the level of the factors.

How to do it [50.18].

Step 1: Choose a CTQ response to be studied by experimentation.
Step 2: Determine the critical parameter to be modified with the experiments. Focus on the significant factors that affect the response.
Step 3: Select the measurement system used to analyze the parameters.
Step 4: Create the transfer function from the experimental data. The transfer function is a mathematical description of the behavior of the system that can be used to create surface plots and optimize the system's performance.
Step 5: Plot the response surface maps to observe the system behavior.
Step 6: Final output: a surface map and an equation that is used to determine the level of the factors. Sensitivity of the factors can also be analyzed.

Inferential Statistics

Inferential statistics are often employed in the *verification* phase.

Purpose. Identify and control variation in the critical responses.

Input. The new product/service's performance data.

Output. The decision on which factors have an effect on the design's response.

Hypotheses and risk: There are null hypothesis and alternate hypothesis. Once we have data, we can determine whether we should accept or reject the null hypothesis, by calculating a test statistic.

The t-test: Used to compare two samples, or one sample with a standard. The null hypothesis is that the means are equal and the difference between the two population means is zero.

Analysis of variance (ANOVA): We use ANOVA when there are more than two samples to compare. ANOVA is used to test whether the means of many samples differ.

H. Statistical Process Control [50.5]

Purpose. Monitor the critical response of the new product/service in the *verify* phase to assess stability and predictability and detect important changes.

Input. The new product/service's performance data.

Output. Assessment of the new product/service's stability, predictability, sigma level and capability for commercialization readiness.

Main considerations. Sample size considerations – sample size should be large enough to provide good sensitivity in detecting out-of-control conditions

Sampling frequency – sampling should be frequent enough to ensure opportunities for process control and improvement.

Concepts. A production/service process that is operating with only chance causes (common causes) of variation present is said to be *in statistical control*. A process is out of control if there exists assignable causes (special causes) that are not part of the chance cause pattern such as improperly adjusted or controlled machines, operator errors, or defective raw material [50.5]. An SPC chart is used to distinguish these two types of causes by upper and lower control limits (UCL and LCL). As long as all the sample points plot within the control limits, the process is assumed to be in statistical control. If a charting point is out of the control limits, this implies that there is evidence that the process is out of control. We then should investigate the assignable causes and take corrective actions.

We can use SPC charts to determine if a new product/service's CTQ measure is in control. If it is, the product/service may move to the mass-production phase.

How to do it [50.5].

Step 1: Select the environment in which the data will be collected.
Step 2: Select the responses and parameters that will be monitored.
Step 3: Select the measurement systems used to acquire the data.
Step 4: Run the system in the prescribed environment

and acquire the data.

Step 5: Plot the data using the appropriate type of SPC chart.

Step 6: Assess the plots for stability and predictability.

Step 7: Calculate the estimates of sigma level and process capability.

50.3 Six Sigma Case Study

In this section, a case study on printed circuit board (PCB) improvement by the author is used to demonstrate the application of Six Sigma, which is digested from *Tong* et al. [50.2] where a more detailed report of this case may be found. This study was conducted in reference to the DMAIC approach, and the objective is to improve the sigma level for a series of products called PSSD in the screening process.

50.3.1 Process Background

This case study was conducted in an electronic company, which is located in an industrial park in southern China. The company manufactures multilayer PCBs by using the surface-mount technology (SMT), which is a technique for placing surface-mount devices (SMDs) on the surface of a PCB. SMDs are micro-miniature leaded or leadless electronic components that are soldered directly to the pads on the surface of the PCB. The major manufacturing processes in the company are solder screen, component placement, and solder reflow. As any defect in any of the solder joints can lead to the failure of the circuit, the screening process is regarded as the most critical process in PCB manufacturing.

The screening process is a manufacturing process that transfers solder paste onto the solder pad of a PCB. The application method for solder paste is printing, and the printing technique used is off-contact printing, in which there is a snap-off distance between a stencil and a PCB. The type of screening machine used to manufacture PSSD products is semiautomatic. During a printing process, two PCBs are placed side-by-side on the holder of a screening machine. The solder paste is then placed onto a stencil manually before printing. The front/back blade makes a line contact with the stencil and a close contact with the given amount of solder paste. The solder paste is then rolled in front of the front/back blade. In this way, solder paste is pressed against the stencil and transferred onto the solder pad through the stencil openings. More detailed operation of a screening process is described in *Tong* et al. [50.2].

50.3.2 Define Phase

In this case, we specifically focus on the improvement of the sigma level of the PCB screening process. In the screening process, the solder paste volume (height) transferred onto the PCB is the most important factor that needs to be controlled carefully. This is because too little solder paste can cause open circuits and too much solder paste can cause bridging between solder pads in the subsequent processes. As a result, the solder paste height on the solder pads is identified as a critical-to-quality (CTQ) characteristic (i. e., the y) that needs to be controlled in a very precise way by the company. According to that, a project team is formed and a project charter is constructed.

50.3.3 Measure Phase

To control the screening process, the project team in the company has asked operators to measure the solder paste height for the PSSD product on five different points on a PCB. The solder paste height on the five points is measured by using the Cyberoptics Cybersentry system every four hours. The gage repeatability and reproducibility (R&R) of the measurement system was verified before the study on the solder paste height is conducted. The gage R&R results ensured that the data from the current measurement system are accurate enough for the following analysis.

50.3.4 Analyze Phase

Currently, six semiautomatic screening machines are used to manufacture the PSSD product. Therefore, the data on solder paste height of these six machines was collected from the company, and the process capability analysis was conducted for these screening machines in order to analyze the current printing performance. According to the analytical results, the process capability in machine number 12 was not satisfactory because the capability index Cp was only 1.021, which was smaller than 1.33 (the four-sigma level). Moreover, another capability index Cpk was 0.387. This showed that

the screening process was off-center. Based on the process capability study, we concluded that there exist both a high variance and a mean shift in the solder paste process. Therefore, we list all the potential factors (xs) that may cause this through brainstorming and constructing a cause and effect diagram.

50.3.5 Improve Phase

In the analysis of the current printing performance, the result showed that the screening process capability of machine number 12 was not satisfactory. After brainstorming with the mechanical engineers in the company, the designed experiments were decided to conduct on machine number 12 in order to determine the optimal settings of all the input factors (xs) in the screening process. In this phase, DOE was used as a core statistical tool for the sigma level improvement.

In the initial experiments, several possible factors that might have influence the printing performance were taken into account. These experiments were used to screen out new factors that have influence on the printing performance. These significant factors would then be included together with the already-known significant factors (solder paste viscosity, speed of squeegee, and pressure of squeegee) in the further experiments. The aim of the further experiments was to determine the standard settings of all the significant factors (i. e., the few, vital xs). By using these optimal settings in the screening process, the printing performance could be improved. As a result, the sigma level can also be enhanced significantly. The detailed DOE setting, analysis, and result can be found in *Tong* et al. [50.2].

50.3.6 Control Phase

To sustain the improvement of the sigma level in the screening process, control plans for all the important xs were proposed to the company. For example, both the CTQ y and the vital xs should be monitored by SPC charts over time, so that the solder paste height variation and the sigma level can be controlled and sustained continuously. Also, the financial benefits through the reduction of COPQ were calculated.

The comparison of the printing performance before and after the project was reported in *Tong* et al. [50.2]. After using the optimal settings, the sigma level of the screening process can be improved from 1.162 to 5.924. This shows that a nearly six-sigma performance can be achieved. According to *Harry* and *Schroeder* [50.19], the level of defects per million opportunities (DPMO) would reduce to near 3.4 and the COPQ would be less than one percent of the sales. As a result, after the Six Sigma practice, the COPQ of the process for this company has been significantly reduced.

50.4 Conclusion

As Six Sigma is evolving over time, the advantages and benefits of other business-excellence approaches can still be learned and utilized in future Six Sigma programs. According to *Hahn* [50.20], combining other tools or methodologies and the Six Sigma methodology may be a future trend. For example, combining lean tools with the Six Sigma methodology has become popular during the last few years. And there are expected to be more combinations in the future.

Recently, Six Sigma efforts have been pushed to both the external suppliers and external customers along a supply chain, which has resulted in even larger overall business impacts and cost savings. I have also observed an increasing trend outside the US, where more and more companies in Asia and Europe, including small-to-medium-sized enterprizes, have implemented various stages of Six Sigma deployment and discovered its far-reaching benefits.

References

50.1 R. D. Snee, R. W. Hoerl: *Six Sigma Beyond the Factory Floor: Deployment Strategies for Financial Services, Health Care, and the Rest of the Real Economy* (Pearson Prentice Hall, Upper Saddle River 2005)

50.2 J. Tong, F. Tsung, B. Yen: A DMAIC approach for printed circuit board quality improvement, Int. J. Adv. Manuf. Technol. **23**, 523–531 (2004)

50.3 T. Y. Ng, F. Tsung, R. H. Y. So, T. S. Li, K. Y. Lam: Six Sigma approach to reducing fall hazards among

cargo handlers working on top of cargo containers: A case study, Int. J. Six Sigma Competitive Advant. **1**(2), 188–209 (2005)
50.4 P. S. Pande, R. P. Neuman, R. R. Cavanagh: *The Six Sigma Way: How GE, Motorola, and Other Top Companies Are Honing Their Performance* (McGraw–Hill, New York 2000)
50.5 D. C. Montgomery: *Introduction to Statistical Quality Control*, 4th edn. (Wiley, New York 2004)
50.6 C. F. J. Wu, M. Hamada: *Experiments: Planning, Analysis, and Parameter Design Optimization* (Wiley, New York 2000)
50.7 D. C. Montgomery: *Design and Analysis of Experiments*, 5th edn. (Wiley, New York 2001)
50.8 B. El-haik, D. M. Roy: *Service Design for Six Sigma: A Road Map for Excellence* (Wiley, New York 2005)
50.9 J. Anton: *Listening to the Voice of the Customer: 16 Steps to a Successful Customer Satisfaction Measurement Program* (Customer Service Group, New York 2005)
50.10 J. Kawakita: *KJ Method: A Scientific Approach to Problem Solving* (Kawakita Research Institute, Tokyo 1975)
50.11 J. M. Spool: The KJ-Technique: A Group Process for Establishing Priorities. Research report. User Interface Engineering, March 2004
50.12 L. Cohen: *Quality Function Deployment: How to Make QFD Work for You* (Addision–Wesley Longman, Reading 1995)
50.13 S. Pugh: *Total Design* (Addison–Wesley, Reading 1995)
50.14 G. S. Altshuller: *Creativity as an Exact Science* (Gordon & Breach, New York 1988)
50.15 C. M. Creveling, J. L. Slutsky, D. Antis Jr.: *Design for Six Sigma in Technology and Product Development* (Prentice Hall PTR, Indianapolis 2003)
50.16 D. H. Stamatis: *Failure Mode and Effect Analysis* (ASQC Quality Press, Milwaukee 1995)
50.17 R. H. Myers, D. C. Montgomery: *Response Surface Methodology* (Wiley Interscience, New York 1995)
50.18 M. J. Anderson, P. J. Whitcomb: *RSM Simplified: Optimizing Processes Using Response Surface Methods for Design of Experiments* (Productivity, Inc., New York 2004)
50.19 M. Harry, R. Schroeder: *Six Sigma* (Doubleday, New York 2000)
50.20 G. J. Hahn: The future of Six Sigma, ASQ Six Sigma Forum Mag. **3**(3), 32–33 (2004)

51. Multivariate Modeling with Copulas and Engineering Applications

This chapter reviews multivariate modeling with copulas and provides novel applications in engineering. A copula separates the dependence structure of a multivariate distribution from its marginal distributions. Properties and statistical inferences of copula-based multivariate models are discussed in detail. Applications in engineering are illustrated via examples of bivariate process control and degradation analysis, using existing data in the literature. A software package has been developed to promote the development and application of copula-based methods.

Section 51.1 introduces the concept of copulas and its connection to multivariate distributions. The most important result about copulas is Sklar's theorem, which shows that any continuous multivariate distribution has a canonical representation by a unique copula and all its marginal distributions. A general algorithm to simulate random vectors from a copula is also presented.

Section 51.2 introduces two commonly used classes of copulas: elliptical copulas and Archimedean copulas. Simulation algorithms are also presented.

Section 51.3 presents the maximum-likelihood inference of copula-based multivariate distributions given the data. Three likelihood approaches are introduced. The exact maximum-likelihood approach estimates the marginal and copula parameters simultaneously by maximizing the exact parametric likelihood. The inference functions for margins approach is a two-step approach, which estimates the marginal parameters separately for each margin in a first step, and then estimates the copula parameters given the the marginal parameters.

The canonical maximum-likelihood approach is for copula parameters only, using uniform pseudo-observations obtained from transforming all the margins by their empirical distribution functions.

Section 51.4 presents two novel engineering applications. The first example is a bivariate process-control problem, where the marginal normality seems appropriate but joint normality is suspicious. A Clayton copula provides a better fit to the data than a normal copula. Through simulation, the upper control limit of Hotelling's T^2 chart based on normality is shown to be misleading when the true copula is a Clayton copula. The second example is a degradation analysis, where all the margins are skewed and heavy-tailed. A multivariate gamma distribution with normal copula fits the data much better than a multivariate normal distribution.

Section 51.5 concludes and points to references about other aspects of copula-based multivariate modeling that are not discussed in this chapter.

An open-source software package for the R project has been developed to promote copula-related methodology development and applications. An introduction to the package and illustrations are provided in the Appendix.

Multivariate methods are needed wherever independence cannot be assumed among the variables under investigation. Multivariate data are encountered in real life much more often than univariate data. This is especially true nowadays with the rapid growth of data-acquisition technology. For example, a quality-control engineer may have simultaneous surveillance of several related quality characteristics or process variables; a reliability analyst may measure the amount of degradation for a certain product repeatedly over time. Because of the dependence among the multiple quality characteristics and repeated measurements, univariate methods are invalid or inefficient. Multivariate methods that can account for the multivariate dependence are needed.

Classic multivariate statistical methods are based on the multivariate normal distribution. Under multivariate normality, an elegant set of multivariate techniques, such as principle-component analysis and factor analysis, has become standard tools and been successful in a variety of application fields. These methods have become so popular that they are often applied without a careful check about whether multivariate normality can reasonably be assumed.

In many applications, the multivariate normal assumption may be inappropriate or too strong to be made. Non-normality can occur in different ways. First, the marginal distribution of some variables may not be normal. For instance, in the degradation analysis in Sect. 51.5, the error rates of magnetic-optic disks at all time points are skewed and heavy-tailed, and hence cannot be adequately modeled by normal distributions. Second, even if all the marginal distributions are normal, jointly these variables may not be multivariate normal. For instance, in the bivariate process-control problem in Sect. 51.5, marginal normality seems appropriate but joint normality is suspicious. In both examples, multivariate distributions that are more flexible than the multivariate normal distribution are needed.

Non-normal multivariate distributions constructed from copulas have proved very useful in recent years in many applications. A copula is a multivariate distribution function whose marginals are all uniform over the unit interval. It is well known that any continuous random variable can be transformed to a uniform random variable over the unit interval by its probability integral transformation. Therefore, a copula can be used to *couple* different margins together and construct new multivariate distributions. This method separates a multivariate distribution into two components, all the marginals and a copula, providing a very flexible framework in multivariate modeling. Comprehensive book references on this subject are *Nelsen* [51.1] and *Joe* [51.2]. For widely accessible introductions, see, for example, *Genest* and *MacKay* [51.3] and *Fisher* [51.4].

Copula-based models have gained much attention in various fields. Actuaries have used copulas when modeling dependent mortality and losses [51.5–7]. Financial and risk analysts have used copulas in asset allocation, credit scoring, default risk modeling, derivative pricing, and risk management [51.8–10]. Biostatisticians have used copulas when modeling correlated event times and competing risks [51.11, 12]. The aim of this chapter is to provide a review of multivariate modeling with copulas and to show that it can be extensively used in engineering applications.

The chapter is organized as follows. Section 51.1 presents the formal definition of copulas and the construction of multivariate distribution with copulas. Section 51.2 presents details about two commonly used classes of copulas: elliptical copulas and Archimedean copulas. Section 51.3 presents likelihood-based statistical inferences for copula-based multivariate modeling. Section 51.4 presents two engineering applications: multivariate process control and degradation analysis. Section 51.5 concludes and suggests future research directions. An open-source software package `copula` [51.13] for the `R` project [51.14] has been developed by the author. A brief introduction to the package and illustrations are presented in the Appendix.

51.1 Copulas and Multivariate Distributions

51.1.1 Copulas

Consider a random vector $(U_1, \ldots, U_p)^\top$, where each margin U_i, $i = 1, \ldots, p$, is a uniform random variable over the unit interval. Suppose the joint cumulative distribution function (CDF) of $(U_1, \ldots, U_p)^\top$ is

$$C(u_1, \ldots, u_p) = \Pr(U_1 \le u_1, \ldots, U_p \le u_p) . \quad (51.1)$$

Then, the function C is called a p-dimensional copula. As *Embrechts* et al. [51.9] noted, this definition of a copula masks some of the problems when construct-

ing copulas using other techniques, by not explicitly specifying what properties a function must have to be a multivariate distribution function; for a more rigorous definition, see for example *Nelsen* [51.1]. However, this definition is operational and very intuitive. For example, one immediately obtains with this definition that, for any p-dimensional copula C, $p \geq 3$, each $k \leq p$ margin of C is a k-dimensional copula and that independence leads to a product copula

$$\Pi_p(u_1, \ldots, u_p) = \prod_{i=1}^{p} u_i \,. \tag{51.2}$$

Every continuous multivariate distribution function defines a copula. Consider a continuous random vector $(X_1, \ldots, X_p)^\top$ with joint CDF $F(x_1, \ldots, x_p)$. Let F_i, $i = 1, \ldots, p$, be the marginal CDF of X_i. Then, $U_i = F_i(X_i)$ is a uniform random variable over the unit interval. One can define a copula C as

$$C(u_1, \ldots, u_p) = F\left\{F_1^{-1}(u_1), \ldots, F_p^{-1}(u_p)\right\} \,. \tag{51.3}$$

The elliptical copulas in Sect. 51.2.1 are constructed this way. Another important class of copulas, Archimedean copulas, is constructed differently (Sect. 51.2.2).

A copula (51.1) can be used to construct multivariate distributions with arbitrary margins. Suppose that it is desired that the i-th margin X_i has marginal CDF G_i. A multivariate distribution function G can be defined via a copula C as

$$G(x_1, \ldots, x_p) = C\{G_1(x_1), \ldots, G_p(x_p)\} \,. \tag{51.4}$$

This multivariate distribution will have the desired marginal distributions.

Clearly, there is a close connection between copulas and multivariate distributions. It is natural to investigate the converse of (51.4). That is, for a given multivariate distribution function G, does there always exist a copula C such that (51.4) holds? If so, is this C unique? These problems are solved rigorously by *Sklar*'s [51.15] theorem in the next section.

51.1.2 Copulas to Multivariate Distributions

Sklar's theorem is the most important result about copulas. The bivariate version of the theorem was established by *Sklar* [51.15] almost half a century ago in the probability metrics literature. The proof in the general p-dimensional case is more involved and can be found in *Sklar* [51.16]. A formal statement of the theorem is as follows [51.1].

Theorem 51.1

Let F be a p-dimensional distribution function with margins $F_1, \ldots, F_p$. Then there exists a p-dimensional copula C such that, for all x in the domain of F,

$$F(x_1, \ldots, x_p) = C\{F_1(x_1), \ldots, F_p(x_p)\} \,. \tag{51.5}$$

If $F_1, \ldots, F_p$ are all continuous, the C is unique; otherwise, C is uniquely determined on $\mathrm{Ran}F_1 \times \cdots \times \mathrm{Ran}F_p$, where $\mathrm{Ran}H$ is the range of H. Conversely, if C is a p-dimensional copula and $F_1, \ldots, F_p$ are distribution functions, then the function F defined by (51.5) is a p-dimensional distribution function with marginal distributions $F_1, \ldots, F_p$.

Sklar's theorem ensures that a continuous multivariate distribution can be separated into two components, univariate margins and multivariate dependence, where the dependence structure is represented by a copula. The dependence structure of a multivariate distribution can be analyzed separately from its margins. It is sufficient to study the dependence structure of a multivariate distribution by focusing on its copula.

The probability density function (PDF) of the CDF F in (51.5) can be found from the PDF of C and $F_1, \ldots, F_p$. The PDF c of the copula C in (51.1) is

$$c(u_1, \ldots, u_p) = \frac{\partial^p C(u_1, \ldots, u_p)}{\partial u_1 \ldots \partial u_p} \,. \tag{51.6}$$

When the density c is known, the density f of the multivariate distribution F in (51.5) is

$$f(x_1, \ldots, x_p) = c\left\{F_1(x_1), \ldots, F_p(x_p)\right\} \prod_{i=1}^{p} f_i(x_i) \,, \tag{51.7}$$

where f_i is the density function of the distribution F_i. Expression (51.7) is called the canonical representation of a multivariate PDF. It will be used to construct likelihood for observed data.

51.1.3 Concordance Measures

The copula of two random variables completely determines any dependence measures that are scale-invariant, that is, measures that remain unchanged under monotonically increasing transformations of the random variables. The construction of the multivariate distribution (51.5) implies that the copula function C is invariant

under monotonically increasing transformations of its margins. Therefore, scale-invariant dependence measures can be expressed in terms of the copulas of the random variables.

Concordance measures of dependence are based on a form of dependence known as concordance. The most widely used concordance measures are Kendall's tau and Spearman's rho. Both of them can be defined by introducing a concordance function between two continuous random vectors (X_1, X_2) and (X_1', X_2') with possibly different joint distributions G and H, but with common margins F_1 and F_2. This concordance function Q is defined as

$$Q = \Pr\left\{(X_1 - X_1')(X_2 - X_2') > 0\right\} - \Pr\left\{(X_1 - X_1')(X_2 - X_2') < 0\right\}, \quad (51.8)$$

which is the difference between the probability of concordance and dis-concordance of (X_1, X_2) and (X_1', X_2'). It can be shown that

$$Q = Q(C_G, C_H) = 4\int_0^1\int_0^1 C_G(u, v)\,\mathrm{d}C_H(u, v) - 1, \quad (51.9)$$

where C_G and C_H are the copulas of G and H, respectively.

For a bivariate random vector (X_1, X_2) with copula C, Kendall's tau is defined as $Q(C, C)$, interpreted as the difference between the probability of concordance and dis-concordance of two independent and identically distributed observations. Therefore, we have

$$\tau = 4\int_0^1\int_0^1 C(u_1, u_2)\,\mathrm{d}C(u_1, u_2) - 1, \quad (51.10)$$

where the range of τ can be shown to be $[-1, 1]$. Spearman's rho, on the other hand, is defined as $3Q(C, \Pi)$, where Π is the product copula obtained under independence. That is,

$$\rho = 12\int_0^1\int_0^1 u_1 u_2\,\mathrm{d}C(u_1, u_2) - 3. \quad (51.11)$$

The constant 3 scales this measure into the range of $[-1, 1]$ (see for example *Nelson* [51.1] p.129). Spearman's rho is proportional to the difference between the probability of concordance and dis-concordance of two vectors: both have the same margins, but one has copula C while the other has the product copula Π. It is straightforward to show that Spearman's rho equals Pearson's product-moment correlation coefficient for the probability-integral-transformed variables $U_1 = F_1(X)$ and $U_2 = F_2(Y)$:

$$\rho = 12E(U_1U_2) - 3 = \frac{E(U_1U_2) - 1/4}{1/12} = \frac{E(U_1U_2) - E(U_1)E(U_2)}{\sqrt{\mathrm{Var}(U_1)\mathrm{Var}(U_2)}}. \quad (51.12)$$

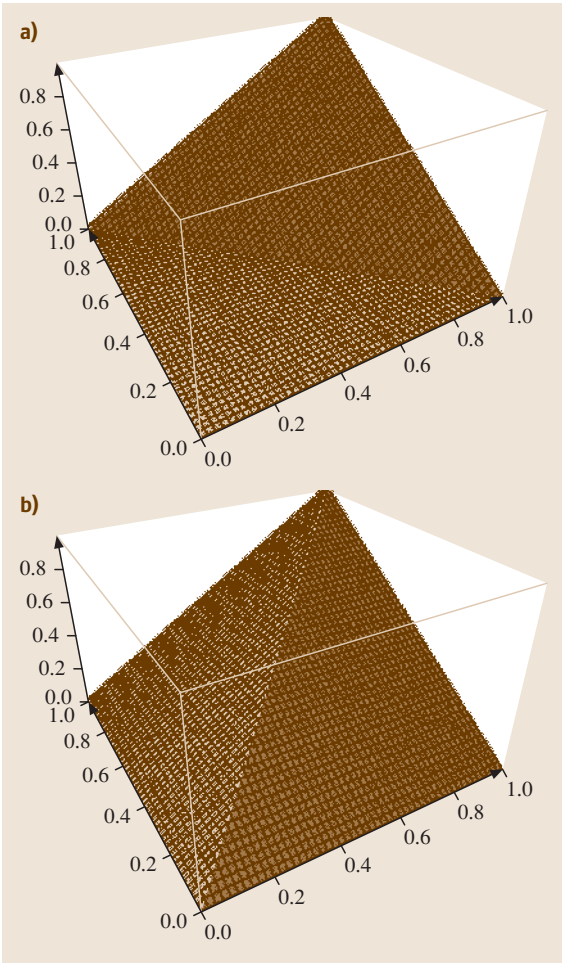


Fig. 51.1a,b Perspective plots of the Fréchet–Hoeffding bounds. **(a)** lower bound; **(b)** upper bound

There are other dependence measures based on copulas. For example, tail dependence is a very important measure when studying the dependence between extreme events. Details can be found in *Joe* [51.2].

51.1.4 Fréchet–Hoeffding Bounds

Important bounds are defined for copulas and multivariate distributions. These bounds are called the Fréchet–Hoeffding bounds, named after the pioneering work of Fréchet and Hoeffding, who independently published their work on this in 1935 and 1940, respectively [51.17]. Define the functions M_p and W_p on $[0,1]^p$ as follows:

$$M_p(u_1,\dots,u_p) = \min(u_1,\dots,u_p),$$
$$W_p(u_1,\dots,u_p) = \max(u_1+\cdots+u_p-n+1,0)\,.$$

Then for every copula C,

$$W_p(u_1,\dots,u_p) \le C(u_1,\dots,u_p) \le M_p(u_1,\dots,u_p)\,. \tag{51.13}$$

These bounds are general bounds, regardless of whether the margins are continuous or not. The function M_p is always a p-dimensional copula for $p \ge 2$. The function W_p fails to be a copula for $p \ge 2$, but it is the best possible lower bound since, for any $u = (u_1,\dots,u_p) \in [0,1]^p$, there exists a copula C (which depends on u) such that $C(u) = W_p(u)$. In the bivariate case, these bounds correspond to perfect negative dependence and perfect positive dependence, respectively. Within a given family of copulas, they may or may not be attained (see for example [51.1] Table 4.1). Figure 51.1 shows the perspective plots of the Fréchet–Hoeffding bounds copulas and the product copula.

Intuitively, perfect dependence should lead to extremes of concordance measures. It can be shown that, for continuous random vector (X_1, X_2) with copula C, $\tau = -1$ (or $\rho = -1$) is equivalent to $C = W_2$ and $\tau = 1$ (or $\rho = 1$) is equivalent to $C = M_2$; see *Embrechts* et al. [51.18] for a proof.

51.1.5 Simulation

Random-number generation from a copula is very important in statistical practice. Consider the p-dimensional copula in (51.1). Let $C_k(u_1,\dots,u_k) = C(u_1,\dots,u_k,1,\dots,1)$ for $k = 2,\dots,p-1$. The conditional CDF of U_k given $U_1 = u_1,\dots,U_{k-1} = u_{k-1}$ is

$$C_k(u_k|u_1,\dots,u_{k-1}) = \frac{\dfrac{\partial^{k-1}C_k(u_1,\dots,u_k)}{\partial u_1\dots\partial u_{k-1}}}{\dfrac{\partial^{k-1}C_{k-1}(u_1,\dots,u_{k-1})}{\partial u_1\dots\partial u_{k-1}}}\,. \tag{51.14}$$

Algorithm (51.1) is a general algorithm to generate a realization $(u_1,\dots,u_p)$ from C via a sequence of conditioning. When the expression for $C_k(\cdot|u_1,\dots,u_{k-1})$ is available, a root-finding routine is generally needed in generating u_k using the inverse CDF method. With realizations from C, one can easily generate realizations from the multivariate distribution (51.4) by applying the inverse CDF method at each margin.

Algorithm 51.1
Generating a random vector from a copula

1. Generate u_1 from a uniform over $[0,1]$.
2. For $k = 2,\dots,p$, generate u_k from $C_k(\cdot|u_1,\dots,u_{k-1})$.

51.2 Some Commonly Used Copulas

We introduce two commonly used copula classes in this section: elliptical copulas and Archimedean copulas. A third class of copulas, extreme-value copulas, is very useful in multivariate extreme-value theory but is omitted here to limit the scope of this chapter; more details about extreme-value copulas can be found in *Joe* [51.2].

51.2.1 Elliptical Copulas

Elliptical copulas are copulas of elliptical distributions. A multivariate elliptical distribution of random vector $(X_1,\dots,X_p)$ centered at zero has density of the form $\phi(t) = \psi(t^\top \Sigma t)$, where $t \in R^p$ and Σ is a $p \times p$ dispersion matrix, which can be parameterized such that $\Sigma_{ij} = \mathrm{Cov}(X_i, X_j)$ [51.19]. Let R_{ij} and τ_{ij} be Pearson's linear correlation coefficient and Kendall's tau between X_i and X_j, respectively. For an elliptical distribution, they are connected through

$$\tau_{ij} = \frac{2}{\pi}\arcsin(R_{ij})\,. \tag{51.15}$$

This relationship makes elliptical copulas very attractive in applications since the similarity between Kendall's tau matrix and the correlation matrix can offer a wide range of dependence structures. Tractable properties similar to those of multivariate normal distributions are another

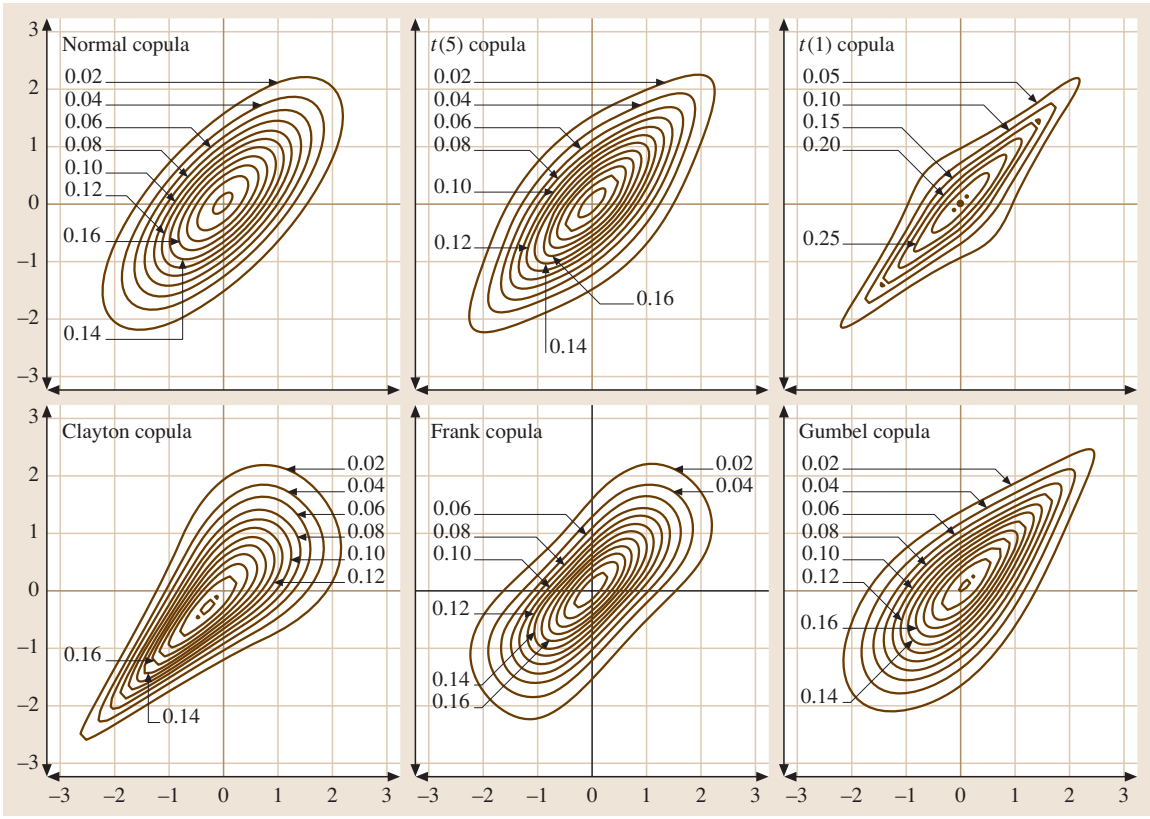


Fig. 51.2 Contours of bivariate distributions with the same marginals but different copulas. Both marginal distributions are standard normal

attractive feature of elliptical copulas. The most popular elliptical distributions are multivariate normal and multivariate t, providing two popular copulas: normal copulas and t copulas.

The normal copula has been widely used in financial applications for its tractable calculus [51.8,20]. Consider the joint CDF Φ_Σ of a multivariate normal distribution with correlation matrix Σ. Let Φ be the CDF of a standard normal variable. A normal copula with dispersion matrix Σ is defined as

$$C(u_1, \ldots, u_p; \Sigma) = \Phi_\Sigma\left[\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_p)\right] . \quad (51.16)$$

The functions Φ, Φ^{-1} and Φ_Σ are available in any reasonably good statistical softwares, which makes their application widely accessible.

The t copula can be constructed similarly [51.21]. Consider the joint CDF $T_{\Sigma,\nu}$ of the standardized multivariate Student's t distribution with correlation matrix Σ and ν degrees of freedom. Let F_{t_ν} be the CDF of the univariate t distribution with ν degrees of freedom. A t copula with dispersion matrix Σ and degrees-of-freedom parameter ν is defined as

$$C(u_1, \ldots, u_p; \Sigma, \nu) = T_{\Sigma,\nu}\left[F_{t_\nu}^{-1}(u_1), \ldots, F_{t_\nu}^{-1}(u_p)\right] . \quad (51.17)$$

These copulas can be used to construct multivariate distributions using (51.5). Note that a normal copula with normal margins is the same as a multivariate normal distribution. However, a t copula with t margins is not necessarily a multivariate t distribution. A multivariate t distribution must have the same degrees of freedom at all the margins. In contrast, a t copula with t margins can have different degrees of freedom at different margins. It offers a lot more flexibility in modeling multivariate heavy-tailed data.

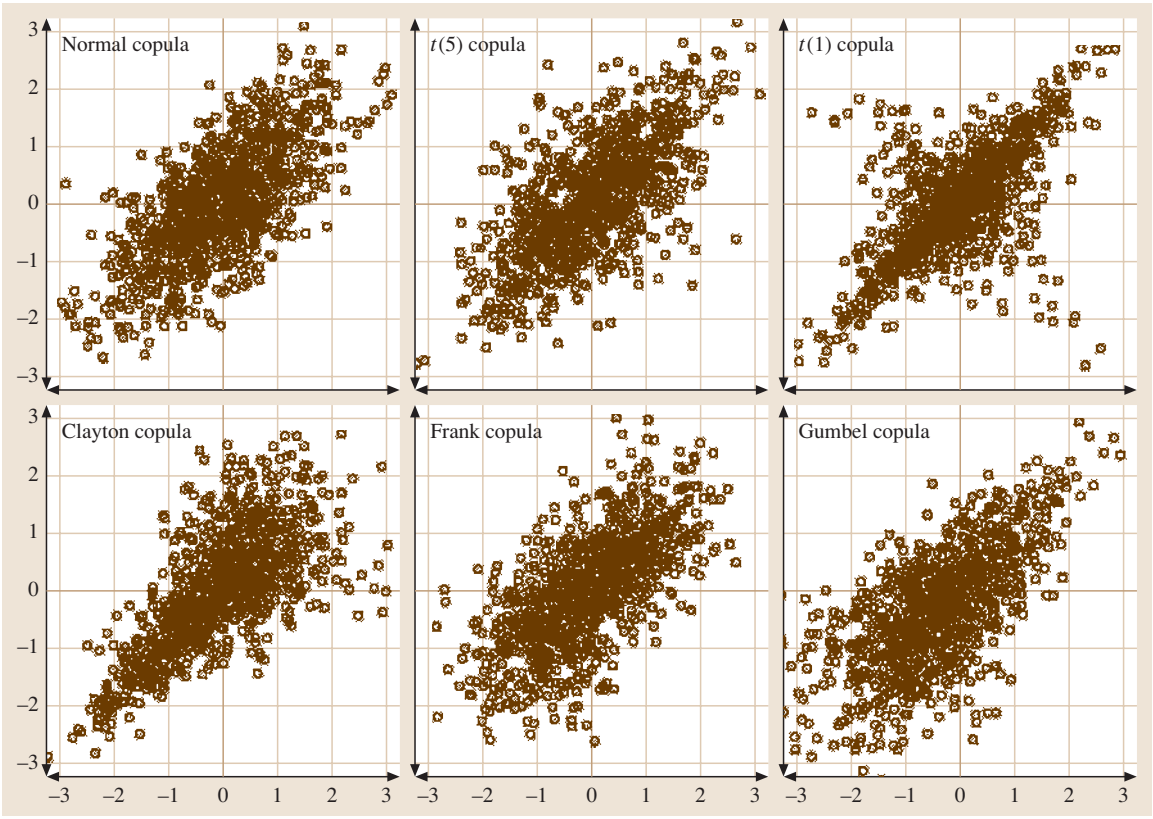


Fig. 51.3 1000 random points from bivariate distributions with the same marginals but different copulas. Both marginal distributions are standard normal

Figure 51.2 shows the density contours of bivariate distributions with the same margins but different copulas. These distributions all have standard normal as both margins, and their values of Kendall's tau are all 0.5. The three plots in the first row of Fig. 51.2 are for a normal copula, t copula with five degrees of freedom, and t copula with one degree of freedom (or Cauchy copula). These densities are computed with (51.7). Note that a normal copula can be viewed as a t copula with infinite degrees of freedom. Figure 51.2 illustrates that the dependence in the tails gets stronger as the number of degrees of freedom decreases.

Simulation from normal copulas and t copulas are straightforward if random-number generators for multivariate normal and t distributions are available. In `R`, the package `mvtnorm` [51.22] provides CDF, PDF and random-number generation for multivariate normal and multivariate t distributions. These facilities are used in the implementation of the package `copula` [51.13].

Figure 51.3 shows 1000 points from the corresponding bivariate distributions in Fig. 51.2.

51.2.2 Archimedean Copulas

Archimedean copulas are constructed via a completely different route without referring to distribution functions or random variables. A key component in this way of construction is a complete monotonic function. A function $g(t)$ is completely monotonic on an interval J if it is continuous there and has derivatives of all orders which alternate in sign, that is,

$$(-1)^k \frac{\mathrm{d}}{\mathrm{d}t^k}\varphi(t) \geq 0, \quad k = 1, 2, \cdots, \tag{51.18}$$

for all t in the interior of J. Let φ be a continuous strictly decreasing function form $[0, 1]$ to $[0, \infty]$ such that $\varphi(0) = \infty$ and $\varphi(1) = 0$, and let φ^{-1} be the inverse of φ. A function defined by

$$C(u_1, \ldots, u_p) = \varphi^{-1}\left[\varphi(u_1) + \cdots + \varphi(u_p)\right] \tag{51.19}$$

Table 51.1 Some one-parameter (α) Archimedean copulas

Family	Generator $\varphi(t)$	Frailty distribution	Laplace transformation of frailty $\mathcal{L}(s) = \varphi^{-1}(s)$
Clayton	$t^{-\alpha} - 1$	Gamma	$(1+s)^{-1/\alpha}$
Frank	$\ln \frac{e^{\alpha t}-1}{e^{\alpha}-1}$	Log series	$\alpha^{-1} \ln [1 + e^{s}(e^{\alpha} - 1)]$
Gumbel	$(-\ln t)^{\alpha}$	Positive stable	$\exp(-s^{1/\alpha})$

is a p-dimensional copula for all $p \geq 2$ if and only if φ^{-1} is completely monotonic over $[0, \infty)$; (see for example [51.1]). The copula C in (51.19) is called an Archimedean copula. The name Archimedean for these copulas comes from a property of the unit cube and copula C which is an analog of the Archimedean axiom for positive real numbers (see [51.1] p. 98 for more details). The function φ is called the generator of the copula. A generator uniquely (up to a scalar multiple) determines an Archimedean copula.

In the bivariate case, an Archimedean copula may be obtained with weaker conditions on the generator φ and its pseudo-inverse $\varphi^{[-1]}$:

$$C(u_1, u_2) = \max \left\{ \varphi^{[-1]} \left[\varphi(u_1) + \varphi(u_2) \right], 0 \right\} , \tag{51.20}$$

where the generator φ is a function with two continuous derivatives such that $\varphi(1) = 0$, $\varphi'(u) < 0$, and $\varphi''(u) > 0$ for all $u \in [0, 1]$, and $\varphi^{[-1]}$ is the pseudo-inverse of φ defined as

$$\varphi^{[-1]}(v) = \begin{cases} \varphi^{-1}(v) & 0 \leq v \leq \varphi(0) , \\ 0 & \varphi(0) \leq v \leq \infty . \end{cases}$$

The generator φ is called a strict generator if $\varphi(0) = \infty$, in which case $\varphi^{[-1]} = \varphi$. *Genest* and *McKay* [51.3] give proofs for some basic properties of bivariate copulas.

The generator φ plays an important role in the properties of an Archimedean copulas. It can be shown that Kendall's tau for an Archimedean copula with generator φ is

$$\tau = 4 \int_0^1 \int_0^1 \frac{\varphi(v)}{\varphi'(v)} \, \mathrm{d}v + 1 . \tag{51.21}$$

This relationship can be used to construct estimating equations that equate the sample Kendall's tau to the theoretical value from the assumed parametric copula family.

Due to the exchangeable structure in (51.19), the associations among all the variables are exchangeable too. As a consequence, an Archimedean copula cannot accommodate negative association unless $p = 2$. For Archimedean copulas with positive associations, there is a mixture representation due to *Marshall* and *Olkin* [51.23]. Suppose that, conditional on a positive latent random variable called the frailty, γ, the distribution of U_i is $F_i(U_i|\gamma) = U_i^{\gamma}$, $i = 1, \ldots, p$, and $U_1, \ldots, U_p$ are independent. Then the copula C of $U_1, \ldots, U_p$ is

$$C(u_1, \ldots, u_p) = E \left(\prod_{i=1}^{p} u_i^{\gamma} \right) , \tag{51.22}$$

where the expectation is taken with respect to the distribution of γ, F_{γ}. Recall that the Laplace transform of γ is

$$\mathcal{L}(s) = E\gamma(\mathrm{e}^{-s\gamma}) = \int_0^{\infty} \mathrm{e}^{-sx} \, \mathrm{d}F_{\gamma}(x) .$$

The Laplace transform has a well-defined inverse $\mathcal{L}^{-1}$. *Marshall* and *Olkin* [51.23] show that the copula in (51.22) is

$$C(u_1, \ldots, u_p) = \mathcal{L} \left[\mathcal{L}^{-1}(u_1) + \cdots + \mathcal{L}^{-1}(u_p) \right] . \tag{51.23}$$

This result suggests that an Archimedean copula can be constructed using the inverse of a Laplace transform as the generator.

Table 51.1 summarizes three commonly used one-parameter Archimedean copulas. A comprehensive list of one-parameter bivariate Archimedean copulas and their properties can be found in Table 4.1 of *Nelson* [51.1]. The three copulas in Table 51.1 all have inverse transforms of some positive random variables as their generators. The Clayton copula was introduced by *Clayton* [51.24] when modeling correlated survival times with a gamma frailty. The Frank copula first appeared in *Frank* [51.25]. It can be shown that the inverse of its generator is the Laplace transform of a log series random variables defined on positive integers. The Gumbel copula can be traced back to *Gumbel* [51.26]. *Hougaard* [51.27] uses a positive stable random variable to derive the multivariate distribution based on a Gumbel copula.

Density contours of bivariate distributions constructed from these three Archimedean copulas are presented in the second row of Fig. 51.2. Both margins of these distributions are still standard normals. The parameters of these copulas are chosen such that the value of Kendall's tau is 0.5. The density of an Archimedean copula can be found by differentiating the copula as in (51.6). When the dimension p is high, the differentiation procedure can be tedious. Symbolic calculus softwares can be used for this purpose. The package `copula` uses the simple symbolic derivative facility in `R` combined with some programming to construct PDF expressions for copulas given the generator function and its inverse function. From Fig. 51.2, one observes that the Frank copula has symmetric dependence. The dependence of the distribution based on the Clayton copula is stronger in the lower-left region than in the upper-right region. In contrast, the dependence of the distribution based on the Gumbel copula is stronger in the upper-right region than in the lower-left region.

Simulation from a general Archimedean can be done using the general Algorithm (51.1) in Sect. 51.2. When the inverse of the generator is known to be the Laplace transform of some positive random variable, an algorithm based on (51.23) is summarized in Algorithm (51.2) [51.6]. This algorithm is very easy to implement, given that a random-number generator of the frailty is available. Gamma-variable generator is available in most softwares. Algorithms for generating positive stable and log series variables can be found in *Chambers* et al. [51.28] and *Kemp* [51.29], respectively. For the bivariate case, the general algorithm (51.1) can be simplified, avoiding numerical root-finding. These algorithms have been implemented in the package `copula` [51.13]. The lower panel of Fig. 51.3 shows 1000 random points generated from the corresponding bivariate distributions with Archimedean copulas in Fig. 51.2.

Algorithm 51.2

tbp Generating a random vector from an Archimedean copula with a known frailty distribution

1. Generate a latent variable γ whose Laplace transformation $\mathcal{L}$ is the inverse generator function φ^{-1}.
2. Generate independent uniform observations $v_1, \ldots, v_p$, $i = 1, \ldots, p$.
3. Output $u_i = \mathcal{L}(-\gamma^{-1} \log v_i)$, $i = 1, \ldots, p$.

51.3 Statistical Inference

This section presents the maximum-likelihood (ML) estimation for multivariate distributions constructed from copulas. Other methods, such as moment methods and nonparametric methods, are less developed for copula-based models and hence omitted.

Suppose that we observe a random sample of size n from a multivariate distribution (51.5):

$$(X_{i1}, \ldots, X_{ip})^\top, \qquad i = 1, \ldots, n\,.$$

The parameter of interest is $\theta = (\beta^\top, \alpha^\top)^\top$, where β is the marginal parameter vector for the marginal distributions $F_i, i = 1, \ldots, p$, and α is the association parameter vector for the copula C. Regression models for the marginal variables can be incorporated easily by assuming that the residuals follow a multivariate distribution (51.5).

51.3.1 Exact Maximum Likelihood

The exact log-likelihood $l(\theta)$ of the parameter vector θ can be expressed from (51.7):

$$l(\theta) = \sum_{i=1}^{n} \log c\left[F_1(X_{i1}; \beta), \ldots, F_p(X_{ip}; \beta); \alpha\right] + \sum_{i=1}^{n}\sum_{j=1}^{p} \log f_i(X_{ij}; \beta)\,. \tag{51.24}$$

The ML estimator of θ is

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta \in \Theta} l(\theta)\,,$$

where Θ is the parameter space.

Under the usual regularity conditions for the asymptotic ML theory, the ML estimator $\hat{\theta}_{\mathrm{ML}}$ is consistent and asymptotically efficient, with limiting distribution

$$\sqrt{n}(\hat{\theta}_{\mathrm{ML}} - \theta_0) \to N\left[0, I^{-1}(\theta_0)\right]\,,$$

where θ_0 is the true parameter value and I is the Fisher information matrix. The asymptotic variance matrix $I^{-1}(\theta_0)$ can be estimated consistently by an empirical variance matrix of the influence functions evaluated at $\hat{\theta}_{\mathrm{ML}}$.

To carry out the ML estimation, one feeds the log-likelihood function $l(\theta)$ to an optimization routine. The asymptotic variance matrix can be obtained from the inverse of an estimated Fisher information matrix, which is the negative Hessian matrix of $l(\theta)$. In R, one constructs the likelihood function using copula densities supplied in the `copula` package, and uses `optim` to maximize it.

The maximization of $l(\theta)$ in (51.24) may be a difficult task, especially when the dimension is high and/or the number of parameters is large. The separation of the margins and copula in (51.24) suggests that one may estimate the marginal parameters and association parameters in two steps, leading to the method in the next subsection.

51.3.2 Inference Functions for Margins (IFM)

The IFM estimation method was proposed by *Joe* and *Xu* [51.30]. This method estimates the marginal parameters β in a first step by

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} \sum_{j=1}^{p} \log f_i(X_{ij}; \beta) \,, \tag{51.25}$$

and then estimates the association parameters α given $\hat{\beta}$ by

$$\hat{\alpha} = \arg\max_{\alpha} \sum_{i=1}^{n} \log c \times \Big[F_1(X_{i1}; \hat{\beta}), \ldots, F_p(X_{ip}; \hat{\beta}); \alpha \Big] \,. \tag{51.26}$$

When each marginal distribution F_i has its own parameters β_i so that $\beta = (\beta_1^\top, \ldots, \beta_p^\top)^\top$, the first step consists of an ML estimation for each margin $j = 1, \ldots, p$:

$$\hat{\beta}_j = \arg\max_{\beta_j} \sum_{i=1}^{n} \log f(X_{ij}; \beta_j) \,. \tag{51.27}$$

In this case, each maximization task has a very small number of parameters, greatly reducing the computational difficulty. This approach is called the two-stage parametric ML method by *Shih* and *Louis* [51.31] in a censored data setting.

The IFM estimator from (51.25) and (51.26), $\hat{\theta}_{\mathrm{IFM}}$, is in general different from the ML estimate $\hat{\theta}_{\mathrm{ML}}$. The limiting distribution of $\hat{\theta}_{\mathrm{IFM}}$ is

$$\sqrt{n}(\hat{\theta}_{\mathrm{IFM}} - \theta_0) \to N\Big[0, G^{-1}(\theta_0)\Big] \,,$$

where G is the Godambe information matrix [51.32]. This matrix has a sandwich form like the usual robust estimation with estimating functions. Detailed expressions can be found in *Joe* [51.2].

Compared to the ML estimator, the IFM estimator has advantages in numerical computations and is asymptotically efficient. Even in finite samples, it is highly efficient relative to the exact ML estimator [51.2]. The IFM estimate can be used as a starting value in an exact ML estimation.

51.3.3 Canonical Maximum Likelihood (CML)

When the association is of explicit interest, the parameter α can be estimated with the CML method without specifying the marginal distribution. This approach uses the empirical CDF of each marginal distribution to transform the observations $(X_{i1}, \ldots, X_{ip})^\top$ into pseudo-observations with uniform margins $(U_{i1}, \ldots, U_{ip})^\top$ and then estimates α as

$$\hat{\alpha}_{\mathrm{CML}} = \arg\max_{\alpha} \sum_{i=1}^{n} \log c(U_{i1}, \ldots, U_{ip}; \alpha) \,. \tag{51.28}$$

The CML estimator $\hat{\alpha}_{\mathrm{CML}}$ is consistent, asymptotically normal, and fully efficient at independence [51.31, 33].

51.4 Engineering Applications

Two engineering applications of copulas are considered in this section: multivariate process control and degradation analysis. An important third application is the modeling of multivariate failure times, which may be censored. We focus on complete-data applications in this chapter. In the example of multivariate process control, marginal normality seems appropriate but joint normality is suspicious. In the example of degradation analysis, the margins are right-skewed and have long tails. We use a gamma distribution for each margin and a normal copula for the association.

51.4.1 Multivariate Process Control

In quality management, multiple process characteristics necessitate a multivariate method for process

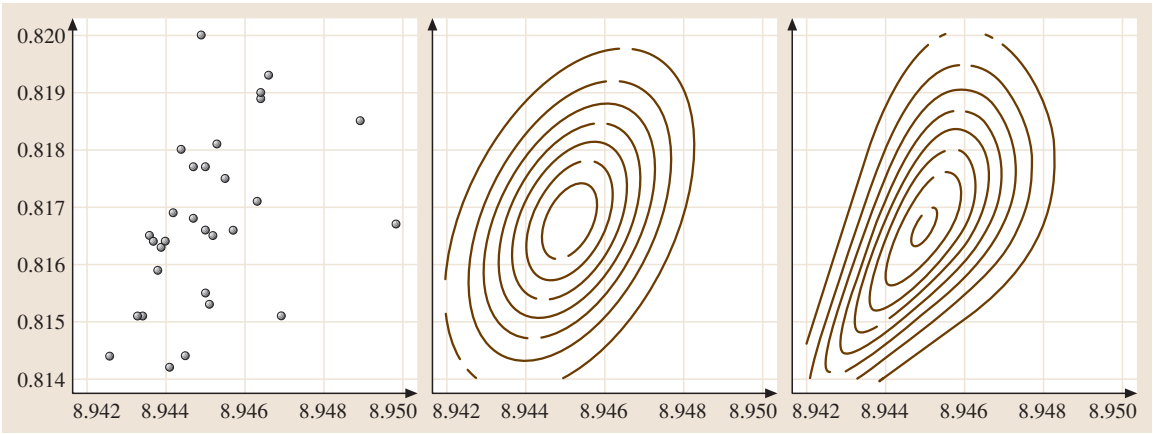


Fig. 51.4 Bivariate process characteristics and parametric fits. *Left*: scatter plot of the data; *center*: contours of bivariate normal fit; *right*: contours of bivariate fit with normal margins and Clayton copula

control. There are three major control charts used in practice: Hotelling's T^2, multivariate cumulative sum (MCUSUM), and multivariate exponentially weighted moving average (MEWMA); see *Lowry* and *Montgomery* [51.34] for a review. The most popular multivariate control chart is the T^2 chart, which has a long history since *Hotelling* [51.35]. *Mason* and *Young* [51.36] give details on how to use it in industrial applications. This method assumes that the multiple characteristics under surveillance are jointly normally distributed. The control limit of the chart is based on the sampling distribution of the statistic T^2, which can be shown to have an F distribution. When the multivariate normal assumption does not hold, due to either univariate or multivariate non-normality, the T^2 control chart based on multivariate normality can be inaccurate and misleading.

Copula-based multivariate distributions open a new avenue for the statistical methods of multivariate process control. The parametric form of the multivariate distribution can be determined from a large amount of historical in-control data. Given a sample of observations when the process is in-control, one can estimate the parameters and propose a statistic that measures the deviation from the target. The exact distribution of this statistic is generally unknown, and the control limit needs to be obtained from bootstrap; see for example *Liu* and *Tang* [51.37].

As an illustration, consider the example of bivariate process control in *Lu* and *Rudy* [51.38]. The data consists of 30 pairs of bivariate measurements from an exhaust manifold used on a Chrysler 5.21 engine in a given model year. They were collected from a machine capability study performed on the machine builder's floor. The sample correlation coefficient is 0.44. The left panel of Fig. 51.4 shows the scatter plot of the 30 observations. The assumption of normality for each margin seems fine from the normal Q–Q plots (not shown). However, the joint distribution may not be a bivariate normal. The scatter plot suggests that the association may be stronger in the lower end than in the higher end of the data. This nonsymmetric association cannot be captured by a symmetric copula, such as those elliptical copula and Frank copula in Fig. 51.2. A better fit of the data may be obtained from a Clayton copula, which allows the bivariate dependence to be stronger in the left tail than in the right tail. The center panel of Fig. 51.4 shows the contours of the ML bivariate normal fit. The right panel of Fig. 51.4 shows the contours of the ML bivariate fit with normal margins and the Clayton copula. The maximized log-likelihood of the two models are 307.64 and 309.87, respectively. A formal test of the difference, which is beyond the scope of this chapter, can be done by comparing non-nested models without knowing the true model based on Kullback–Leibler information [51.39].

The T^2 control chart of *Lu* and *Rudy* [51.38] is a phase II chart for single observations to detect any departure of the underlying process from the standard values. Suppose that we observe a random sample of p-dimensional multivariate observations with sample size m. Let $\bar{X}_m$ and S_m be the sample mean vector and sample covariance matrix, respectively. For a future p-dimensional multivariate observation X, the T^2 is defined as

$$T^2 = (X - \bar{X}_m)^\top S_m^{-1} (X - \bar{X}) \,. \tag{51.29}$$

Table 51.2 Comparison of T^2 percentiles when the true copula is normal and when the true copula is Clayton with various Kendall's τ. The percentiles under Clayton copulas are obtained from 100 000 simulations

Percentiles	Normal copula	Clayton copula			
		$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.6$	$\tau = 0.8$
90%	5.357	5.373	5.416	5.590	5.868
95%	7.150	7.253	7.468	8.061	9.396
99%	11.672	12.220	13.080	15.764	23.526
99.73%	15.754	16.821	18.611	24.173	41.123

Under joint normality, it can be shown that the exact distribution of

$$\frac{m^2 - mp}{p(m+1)(m-1)} T^2$$

is F with degrees of freedom p and $m-p$. The exact upper control limit (UCL) for T^2 with level α is then

$$\mathrm{UCL}_\alpha = \frac{p(m+1)(m-1)}{m^2 - mp} F_{1-\alpha;\,p,m-p}\,, \qquad (51.30)$$

where $F_{1-\alpha;\,p,m-p}$ is the $100(1-\alpha)$ percentile of an F distribution with p and $m-p$ degrees of freedom. In this example, $m = 30$, $p = 2$. The exact upper control limit for T^2 with level α is then

$$\begin{aligned}\mathrm{UCL}_\alpha &= 2(30+1)(30-1)/[30^2 - 2(30)]F_{1-\alpha;2,28} \\ &= 2.14 F_{1-\alpha;2,28}\,.\end{aligned}$$

With $\alpha = 0.9973$, the control limit UCL = 15.75.

When the true copula is a Clayton copula but is misspecified as a normal copula, the control limit in (51.30) can be inaccurate and hence misleading. By comparing the contours of a normal copula model with those of a Clayton copula model in Fig. 51.2, one can conjecture that, if the true copula is a Clayton copula, then $\Pr(T^2 > \mathrm{UCL}_\alpha)$ will be greater than its nominal level α, because the bivariate density with the Clayton copula is more concentrated on the lower-left part of the plot than the bivariate normal density. In other words, in order to maintain the control level α, one needs to increase the UCL of the T^2 chart. This difference obviously depends on the sample size m and the association parameter of the true Clayton copula. For a given sample size m and a Kendall's τ value, which determines the association strength of a Clayton copula, the control limit of T^2 can be obtained by simulation. Table 51.2 compares the 90%, 95%, 99%, and 99.73% percentiles of T^2 when the true copula is normal and when the true copula is Clayton. The percentiles under Clayton copulas are obtained from 100 000 simulations. The true Clayton copulas are parameterized to give Kendall's τ values 0.2, 0.4, 0.6, and 0.8. From Table 51.2, one observes that the simulated percentiles of T^2 are greater than those based on the F distribution under the normal assumption. The control region based on the normal assumption is smaller than expected, which will result in investigating the process more often than necessary when the process is actually in control. The difference increases with the strength of the association.

This example illustrates that a non-normal joint distribution may have an important influence on the control limit of the widely used T^2 chart, even when both the margins are normals. The T^2 statistic still measures the deviance from the target, but its distribution is unknown under the non-normal model. A comprehensive investigation of multivariate process control using copula is a future research direction.

51.4.2 Degradation Analysis

Performance degradation data has repeated measures over time for each test unit (see for example *Meeker* and *Escobar* [51.40] Chap. 13). These repeated measures on the same unit are correlated. There is a voluminous statistical literature on the analysis of repeated mea-

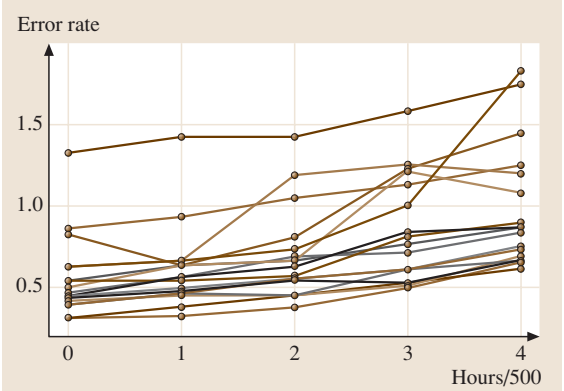


Fig. 51.5 Error rates ($\times 10^5$) of 16 magneto-optic data-storage disks measured every 500 h

Table 51.3 IFM fit for all the margins using normal and gamma distributions, both parameterized by mean and standard deviation. Presented results are log-likelihood (Loglik), estimated mean, and estimated standard deviation (StdDev) for each margin under each model

Time in units of 500 h	Normal margins			Gamma margins		
	Loglik	Mean	StdDev	Loglik	Mean	StdDev
0	−0.484	0.565	0.062	2.568	0.565	0.054
1	−0.526	0.617	0.063	2.538	0.617	0.054
2	−2.271	0.709	0.070	−0.125	0.709	0.064
3	−4.441	0.870	0.080	−3.269	0.870	0.078
4	−6.996	1.012	0.094	−5.205	1.012	0.087

surements (see, for example, *Davis* [51.41]. Analysis of such data has been implemented in popular statistical softwares, for example, `PROC MIXED` of the SAS system [51.42] and the `nlme` package [51.43] for `R` and `Splus`. Continuous response variables are generally assumed to be normally distributed and a multivariate normal distribution is used in likelihood-based approaches. The following example shows that a multivariate gamma distribution with normal copula can provide a much better fit to the data than a multivariate normal distribution.

Degradation data on block error rates of 16 magneto-optic data storage disks are collected every 500 h for 2000 h at 80 °C and 85% relative humidity [51.44]. Figure 51.5 shows these error rates at all five time points. A degradation analysis often needs to fit a curve for the degradation trend in order to allow predictions at unobserved time points. Before choosing a curve to fit, we first carry out exploratory data analysis using the two-step IFM method to look into parametric modeling for each margin and copula separately.

Separate parametric fits for each margin is the first step of the IFM approach in Sect. 51.4. Two parametric models for each margin are used: normal and gamma. To make the parameters comparable across models, the gamma distribution is parameterized by its mean μ and standard deviation σ, giving a density function of

$$f(x;\mu,\sigma)=\frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}e^{-\frac{x}{\beta}}\,, \tag{51.31}$$

where $\alpha=\mu^2/\sigma^2$ and $\beta=\sigma^2/\mu$. Table 51.3 summarizes the separate parametric fits for each margins using normal and gamma distributions. For all the margins, the gamma distribution fit yields higher log-likelihood than the normal distribution fit. The estimated mean from both models are the same for the first three digits after the decimal point. The estimated standard deviation is noticeably lower in the gamma model, especially at earlier time points where the data are more skewed and heavier-tailed. These estimates are consistent with the descriptive statistics of each time point, suggesting that the mean error rate is increasing over time, and their standard errors is increasing with the mean level.

Given the parametric fit for each margins, we can explore copula fitting in the second step of IFM. Due to the small number of observations, we choose single-parameter normal copulas with three dispersion structures: AR(1), exchangeable, and Toeplitz. In particular, with $p=5$, the dispersion matrices with parameter ρ under these structures are, respectively,

$$\begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \rho^4 \\ \rho & 1 & \rho & \rho^2 & \rho^3 \\ \rho^2 & \rho & 1 & \rho^2 & \rho^2 \\ \rho^3 & \rho^2 & \rho & 1 & \rho \\ \rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho & \rho \\ \rho & \rho & 1 & \rho & \rho \\ \rho & \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{pmatrix}, \text{ and}$$

$$\begin{pmatrix} 1 & \rho & & & \\ \rho & 1 & \rho & & \\ & \rho & 1 & \rho & \\ & & \rho & 1 & \rho \\ & & & \rho & 1 \end{pmatrix}. \tag{51.32}$$

Table 51.4 summarizes the log-likelihood and the estimated association parameter ρ for the given estimated margins in Table 51.3. Note that the log-likelihood values are not comparable across models with different margins because the data being used in the estimation are different. They are comparable when the modeled margins are the same. For both normal margins and gamma margins, the AR(1) structure gives the highest log-likelihood value. The estimated parameter is about 0.9, indicating high dependence among repeated measurements.

Table 51.4 also presents the normal copulas estimation using the CML method. No parametric distribution is assumed for each margin. The empirical distribution is used to transform the observations of each margin

Table 51.4 IFM and CML fit for single-parameter normal copulas with dispersion structures: AR(1), exchangeable, and Toeplitz

Dispersion structure	IFM fit				CML fit	
	Normal margins		Gamma margins		Empirical margins	
	Loglik	$\hat{\rho}$	Loglik	$\hat{\rho}$	Loglik	$\hat{\rho}$
AR(1)	39.954	0.917	66.350	0.892	10.380	0.964
Exchangeable	38.618	0.868	62.627	0.791	9.791	0.942
Toeplitz	23.335	0.544	39.975	0.540	5.957	0.568

into uniform variables in [0, 1], which are then used in (51.28). The CML fit also shows that the AR(1) structure gives the highest log-likelihood and that the within-disk dependence is high. Based on these exploratory analysis, the AR(1) structure is used for the dispersion matrix of normal copula in an exact ML analysis.

We now present the exact ML estimation of a degradation model. For the sake of simplicity, we use a linear function of time to model the mean $\mu(t)$ and a linear function of $\mu(t)$ to model the logarithm of the standard deviation $\sigma(t)$. That is,

$$\mu(t) = \phi_0 + \phi_1 t, \tag{51.33}$$

$$\log \sigma(t) = \psi_0 + \psi_1 [\mu(t) - 1.0] , \tag{51.34}$$

where ϕ_0, ϕ_1, ψ_0, and ψ_1 are parameters, and the function of $\log \sigma(t)$ is centered at 1.0 for easier prediction of the variance at higher error rates. Two parametric models are considered for the repeated error rates: (1) multivariate normal and (2) multivariate gamma via a normal copula. Note that the two models both use the normal copula. The marginal distributions of the two models at time t are both parameterized by mean $\mu(t)$ and standard deviation $\sigma(t)$ for comparison purpose. A similar parameterization was used in *Lambert* and *Vandenhende* [51.45] and *Frees* and *Wang* [51.7].

Table 51.5 summarizes the maximum-likelihood estimate of the parameters and their standard errors for both models. These estimates for both marginal parameters and the copula parameter are virtually the same or very close to each other. However, the standard errors of these estimates are noticeably smaller in the multivariate gamma model. The maximized log-likelihood from the gamma model is much higher than that from the normal model. Given that both models have the same number of parameters, the multivariate gamma distribution fits the data much better.

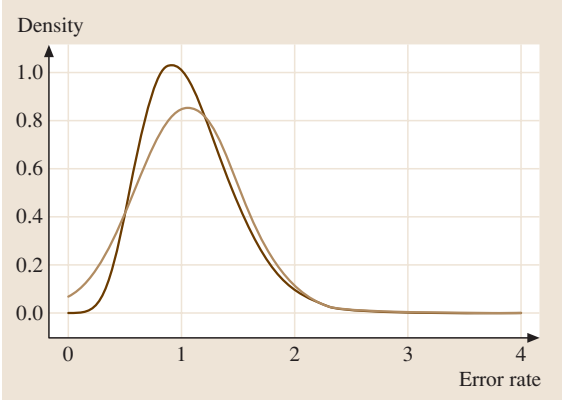


Fig. 51.6 Predictive density of disk error rate at 2500 h. The *dark line* is from the gamma model; the *gray line* is from the normal model

The difference between the two models can also be illustrated by their predictive density of the error rate at 2500 h. Figure 51.6 presents the densities of the error rate at 2500 h using the estimated mean $\mu(2500)$ and

Table 51.5 Maximum-likelihood results for the disk error-rate data. Parameter estimates, standard errors and log-likelihood are provided for both the multivariate normal model and the multivariate gamma model with a normal copula. The second entry of each cell is the corresponding standard error

Model	Marginal parameters				Copula parameter	Loglik
	Mean		StdDev.			
	ϕ_0	ϕ_1	ψ_0	ψ_1	ρ	
Normal	0.564	0.099	−0.849	1.439	0.899	34.719
	0.057	0.019	0.262	0.557	0.034	
Gamma	0.564	0.101	−0.986	1.383	0.900	48.863
	0.051	0.015	0.185	0.442	0.033	

$\sigma(2500)$ obtained with $\hat{\phi}_0$, $\hat{\phi}_1$, $\hat{\psi}_0$, and $\hat{\psi}_1$. The normal model gives mean 1.058 and standard deviation 0.465, while the gamma model gives mean 1.070 and standard deviation 0.411. Although the mean values are close, the gamma model gives a small standard deviation and captures the skewness and long tail of the data.

51.5 Conclusion

This chapter reviews multivariate modeling with copulas and provides novel applications in engineering. Multivariate distribution construction using copulas and their statistical inferences are discussed in detail. Engineering applications are illustrated via examples of bivariate process control and degradation analysis, using existing data in the literature. Copulas offer a flexible modeling strategy that separates the dependence structure from the marginal distributions. Multivariate distributions via copula apply to a much wider range of multivariate scenarios than the traditionally assumed multivariate normal distribution. A publicly available R package has been developed to promote the research on copulas and their applications.

Some important topics about copulas are not discussed in this chapter. The survival function is of great concern in failure-time data analysis. Similarly to (51.5), a multivariate survival function can be constructed via a copula with

$$S(x_1, \ldots, x_p) = C[S_1(x_1), \ldots, S_p(x_p)] ,$$

where S is the joint survival function and $S_i(t) = 1 - F(t)$ is the i-th marginal survival function, $i = 1, \ldots, p$. In this setting C is called a survival copula. Censoring presents an extra difficulty for multivariate failure-time data analysis. *Georges* et al. [51.46] gives an excellent review on multivariate survival modeling with copulas. This chapter has focused on parametric copula models. Standard inferences of the maximum-likelihood method can be applied under the usual regularity conditions. However, which copula to choose and how well it fits the data are important practical problems. Diagnostic tools, particularly graphical tools, can be very useful. There are not many works in this direction; some recent ones are *Wang* and *Wells* [51.11] and *Fermanian* [51.47].

Copulas have had a long history in the probability literature [51.17]. Recent development and application in insurance, finance and biomedical research have been successful. With this chapter, it is hoped to encourage engineering researchers and practitioners to stimulate more advancement on copulas and seek more applications.

51.A Appendix

51.A.1 The R Package Copula

Overview

Software implementation is very important in promoting the development and application of copula-based approaches. Unfortunately, there are few software packages available for copula-based modeling. One exception is the `finmetrics` module [51.48] of `Splus` [51.49]. For an array of commonly used copulas, the `finmetrics` module provides functions to evaluate their CDF and PDF, generate random numbers from them, and fit them for given data. However, these functionalities are limited because only bivariate copulas are implemented. Furthermore, the software is commercial. It is desirable to have an open-source platform for the development of copula methods and applications.

R is a free software environment for statistical computing and graphics [51.14]. It runs on all platforms, including Unix/Linux, Windows, and MacOS. Cutting-edge statistical developments are easily incorporated into R by the mechanism of contributed packages with quality assurance [51.50]. It provides excellent graphics and interfaces easily with lower-level compiled code such as C/C++ or FORTRAN. An active developer–user interaction is available through the R-help mailing list. Therefore, it is a natural choice to write an R package for copulas.

The package `copula` [51.13] is designed using the object-oriented feature of the S language [51.51]. It is publicly available at the Comprehensive R Archive Network [CRAN, `http://www.r-project.org`]. S4 classes are created for elliptical copulas and Archimedean copulas with arbitrary dimension; the extreme-value copula class is still to be implemented at the time of writing. For each copula family, methods of density, distribution, and random-number generator are implemented. For visualization, methods of contour and perspective plots are provided for bivariate copulas.

More facilities, such as extreme-value copulas, association measures and tail dependence measures, will be included in future releases of the package.

Illustration

The package `copula` depends on the contributed packages `mvtnorm`, `scatterplot3d`, and `sn`, taking advantages of the existing facilities in these packages that are relevant. The package needs to be loaded before using:

```
> library(copula)
```

The package is well documented following the requirement of the `R` project [51.50]. A list of help topics can be obtained from:

```
> library(help = copula)
```

We illustrate the features of the package from the following aspects by examples.

Constructing copula objects. An object of class `normalCopula` can be created by

```
> mycop1 <- ellipCopula(family =
"normal", param = c(0.707, 0.5,
0.2), dim = 3, dispstr = "un").
```

The created object `mycop1` is of class `normalCopula`, which inherits `ellipCopula` and `copula`. It has dimension three, with an `unstructured` dispersion matrix

$$\begin{pmatrix} 1.000 & 0.707 & 0.500 \\ 0.707 & 1.000 & 0.200 \\ 0.500 & 0.200 & 1.000 \end{pmatrix}.$$

An object of class `tCopula` can be created similarly, with an extra argument for the degrees of freedom, `df`:

```
> mycop2 <- ellipCopula(family =
"t", param = c(0.9, 0.5, 0.2),
df = 5, dim = 3, dispstr = "un")
```

Examples of objects of Archimedean copulas can be created by:

```
> mycop3 <- archmCopula(family =
"clayton", param = 2, dim = 3)
> mycop4 <- archmCopula(family =
"frank", param = 5.736, dim = 3)
> mycop5 <- archmCopula(family =
"gumbel", param = 2, dim = 3)
```

Constructing Multivariate Distribution via Copulas. An object of multivariate distributions via copulas can be constructed by specifying the copula and its marginal distributions. For example:

```
> mymvd1 <- mvdc(copula =
normalCopula(0.5, dim = 2),
margins = c("norm", "gamma"),
paramMargins = list(list(mean = 0,
sd = 2), list(shape = 2, rate = 2)))
```

The created object `mymvd1` is of class `mvdc`. It is a bivariate distribution constructed via a normal copula. One of the marginal distributions is normal with mean 0 and standard deviation 2. The other marginal distribution is gamma with shape 2 and rate 2.

Density, Distribution, and Simulation. The density and distribution of an object of `copula` class are obtained through the generic method functions `dcopula` and `pcopula`. These functions for an object of the `mvdc` class are obtained through the method functions `dmvdc` and `pmvdc`. The density method `dmvdc` for an `mvdc` object can be used to construct the likelihood for a given dataset.

For Archimedean copulas, obtaining the density function by differentiating the copula can be tedious. The `copula` package provides expressions for the PDF from symbolic calculations. The following code returns the CDF and PDF expressions of a Clayton copula with parameter α:

```
> mycop3@exprdist$cdf
(1 + (u1^(-alpha) - 1 + u2^(-alpha) -
1 + u3^(-alpha) - 1))^(-1/alpha)

> mycop3@exprdist$pdf
(1 + (u1^(-alpha) - 1 + u2^(-alpha) -
1 + u3^(-alpha) - 1))^((((-1/alpha)
- 1) - 1) - 1) * ((((-1/alpha) - 1)
- 1) * (u3^((-alpha) - 1) * (-alpha)))
* (((-1/alpha) - 1) * (u2^((-alpha)
- 1) * (-alpha))) * ((-1/alpha)
* (u1^((-alpha) - 1) * (-alpha)))
```

These can be exported into other programming languages with little or minor modification.

The methods `rcopula` and `rmvdc` generate random numbers from a `copula` or `mvdc` object. The following code generates five observations from `mymvd1` and evaluates the density and distribution at these points:

```
> n <- 5
> x <- rmvdc(mymvd1, n)
> x
          [,1]       [,2]
[1,] -2.7465647 0.6404319
[2,] -1.2674922 0.2707347
[3,] -1.8268522 0.4869647
[4,] 0.2742349 1.1763891
[5,] 2.5947601 1.6410892
```

```
> cbind(dmvdc(mymvd1, x), pmvdc
(mymvd1, x))
           [,1]       [,2]
[1,] 0.06250414 0.06282100
[2,] 0.14514281 0.06221677
[3,] 0.13501126 0.09548434
[4,] 0.10241486 0.45210057
[5,] 0.03698266 0.78582431
```

Bivariate Contour and Perspective Plot.
The `contour` and `persp` methods are implemented for the `copula` and `mvdc` classes. The following code examples draw the contours and perspective plot of the CDF for a bivariate t copula with correlation $\rho = 0.707$:

```
> contour(tCopula(0.707), pcopula)
> persp(tCopula(0.707), pcopula)
```

To draw these plots for an `mvdc` object, the ranges of the margins need to be specified:

```
> persp(mymvd1, dmvdc, xlim =
c(-4, 4), ylim = c(0, 3))
> contour(mymvd1, dmvdc, xlim =
c(-4, 4), ylim = c(0, 3))
```


References

51.1 R. B. Nelsen: *An Introduction to Copulas* (Springer, Berlin Heidelberg New York 1999)
51.2 H. Joe: *Multivariate Models and Dependence Concepts* (Chapman Hall, Norwell 1997)
51.3 C. Genest, J. MacKay: The joy of copulas: Bivariate distributions with uniform marginals (Com: 87V41 P248), Am. Statist. **40**, 280–283 (1986)
51.4 N. I. Fisher: Copulas. In: *Encyclopedia of Statistical Sciences*, ed. by S. Kotz, C. B. Read, D. L. Banks (Wiley, New York 1997) pp. 159–163
51.5 E. W. Frees, J. Carriere, E. A. Valdez: Annuity valuation with dependent mortality, J. Risk Insur. **63**, 229–261 (1996)
51.6 E. W. Frees, E. A. Valdez: Understanding relationships using copulas, North Am. Actuar. J. **2**, 1–25 (1998)
51.7 E. W. Frees, P. Wang: Credibility using copulas, North Am. Actuar. J. **9**, 31–48 (2005)
51.8 E. Bouyè, V. Durrleman, A. Bikeghbali, G. Riboulet, T. Roncalli: *Copulas for Finance – A Reading Guide and Some Applications*, *Working Paper* (Groupe de Recherche Opérationnelle, Crédit Lyonnais, Lyon 2000)
51.9 P. Embrechts, F. Lindskog, A. McNeil: Modelling dependence with copulas and applications to risk management. In: *Handbook of Heavy Tailed Distribution in Finance*, ed. by S. Rachev (Elsevier, Amsterdam 2003) pp. 329–384
51.10 U. Cherubini, E. Luciano, W. Vecchiato: *Copula Methods in Finance* (Wiley, New York 2004)
51.11 W. Wang, M. T. Wells: Model selection and semiparametric inference for bivariate failure-time data (C/R: p73–76), J. Am. Statist. Assoc. **95**, 62–72 (2000)
51.12 G. Escarela, J. F. Carrière: Fitting competing risks with an assumed copula, Statist. Methods Med. Res. **12**, 333–349 (2003)
51.13 J. Yan: *Copula: Multivariate Dependence with Copula, R package version 0.3-3* 2005) CRAN, http://cran.r-project.org
51.14 R Development Core Team: *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna 2005)
51.15 A. W. Sklar: Fonctions de répartition à *n* dimension et leurs marges, Publ. Inst. Statist. Univ. Paris **8**, 229–231 (1959)
51.16 A. Sklar: Random variables, distribution functions, and copulas – A personal look backward and forward. In: *Distributions with Fixed Marginals and Related Topics*, *IMS Lecture Notes Monogr. Ser.*, Vol. 28, ed. by L. Rüschendorf, B. Schweizer, M. D. Taylor (Institute of Mathematical Statistics, Bethesda 1996) pp. 1–14
51.17 B. Schweizer: Thirty years of copulas. In: *Advances in Probability Distributions with Given Margins: Beyond the Copulas*, ed. by G. Dall'Aglio, S. Kotz, G. Salinetti (Kluwer Academic, Dordrecht 1991) pp. 13–50
51.18 P. Embrechts, A. McNeil, D. Straumann: Correlation and dependence in risk management: Properties and pitfalls. In: *Risk Management: Value at Risk and Beyond*, ed. by M. Dempster (Cambridge Univ. Press, Cambridge 2002) pp. 176–223
51.19 K.-T. Fang, S. Kotz, K. W. Ng: *Symmetric Multivariate and Related Distributions* (Chapman Hall, Norwell 1990)
51.20 P. X.-K. Song: Multivariate dispersion models generated from Gaussian copula, Scandin. J. Statist. **27**, 305–320 (2000)
51.21 S. Demarta, A. J. McNeil: The *t* copula and related copulas, Int. Statist. Rev. **73**, 111–129 (2005)
51.22 A. Genz, F. Bretz, T. Hothorn: *Mvtnorm: Multivariate Normal and T Distribution, R package version 0.7-2* 2005) CRAN, http://cran.r-project.org
51.23 A. W. Marshall, I. Olkin: Families of multivariate distributions, J. Am. Statist. Assoc. **83**, 834–841 (1988)
51.24 D. G. Clayton: A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in

chronic disease incidence, Biometrika **65**, 141–152 (1978)
51.25 M.J. Frank: On the simultaneous associativity of $F(x,y)$ and $x+y-F(x,y)$, Aequ. Math. **19**, 194–226 (1979)
51.26 E.J. Gumbel: Bivariate exponential distributions, J. Am. Statist. Assoc. **55**, 698–707 (1960)
51.27 P. Hougaard: A class of multivariate failure time distributions (Corr: V75 p395), Biometrika **73**, 671–678 (1986)
51.28 J.M. Chambers, C.L. Mallows, B.W. Stuck: A method for simulating stable random variables (Corr: V82 P704; V83 P581), J. Am. Statist. Assoc. **71**, 340–344 (1976)
51.29 A.W. Kemp: Efficient generation of logarithmically distributed pseudo-random variables, Appl. Statist. **30**, 249–253 (1981)
51.30 H. Joe, J. Xu: *The Estimation Method of Inference Functions for Margins for Multivariate Models, Tech. Rep. 166* (Department of Statistics, University of British Columbia, Vancouver 1996)
51.31 J.H. Shih, T.A. Louis: Inferences on the association parameter in copula models for bivariate survival data, Biometrics **51**, 1384–1399 (1995)
51.32 V.P. Godambe: An optimum property of regular maximum likelihood estimation (Ack: V32 p1343), Annal. Math. Statist. **31**, 1208–1212 (1960)
51.33 C. Genest, K. Ghoudi, L.-P. Rivest: A semiparametric estimation procedure of dependence parameters in multivariate families of distributions, Biometrika **82**, 543–552 (1995)
51.34 C.A. Lowry, D.C. Montgomery: A review of multivariate control charts, IIE Trans. **27**, 800–810 (1995)
51.35 H. Hotelling: Multivariate quality control – Illustrated by the air testing of sample bombsights. In: *Techniques of Statistical Analysis*, ed. by C. Eisenhart, M.W. Hastay, W.A. Wallis (McGraw–Hill, New York 1947) pp. 111–184
51.36 R.L. Mason, J.C. Young: *Multivariate Statistical Process Control with Industrial Applications*, ed. by ASA-SIAM Ser. Statist. Appl. Probab. (SIAM, Philadelphia 2001) p. 263
51.37 R.Y. Liu, J. Tang: Control charts for dependent and independent measurements based on bootstrap methods, J. Am. Statist. Assoc. **91**, 1694–1700 (1996)
51.38 M.-W. Lu, R.J. Rudy: Multivariate control chart. In: *Recent Advances in Reliability and Quality Engineering*, ed. by H. Pham (World Scientific, Singapore 2001) pp. 61–74
51.39 Q.H. Vuong: Likelihood ratio tests for model selection and non-nested hypotheses (STMA V31 0456), Econometrica **57**, 307–333 (1989)
51.40 W.Q. Meeker, L.A. Escobar: *Statistical Methods for Reliability Data* (Wiley, New York 1998)
51.41 C.S. Davis: *Statistical Methods for the Analysis of Repeated Measurements* (Springer, Berlin Heidelberg New York 2002)
51.42 R.C. Littell, G.A. Milliken, W.W. Stroup, R.D. Wolfinger: *SAS System for Mixed Models* (SAS Institute, Cary 1996)
51.43 J.C. Pinheiro, D.M. Bates: *Mixed-Effects Models in S and S-PLUS* (Springer, Berlin, New York 2000)
51.44 W.P. Murray: Archival life expectancy of 3M magneto-optic media, J. Magn. Soc. Jpn. **17**, 309–314 (1993)
51.45 P. Lambert, F. Vandenhende: A copula-based model for multivariate non-normal longitudinal data: Analysis of a dose titration safety study on a new antidepressant, Statist. Med. **21**, 3197–3217 (2002)
51.46 P. Georges, A.-G. Lamy, E. Nicolas, G. Quibel, T. Roncalli: *Multivariate survival modelling: A Unified Approach with Copulas, Working Paper* (Groupe de Recherche Opérationnelle, Crédit Lyonnais, Lyon 2001)
51.47 J.-D. Fermanian: Goodness-of-fit tests for copulas, J. Multivariate Anal. **95**, 119–152 (2005)
51.48 Insightful Corp.: *S + Finmetrics Reference Manual* (Insightful, Seattle 2002)
51.49 Insightful Corp.: *S-PLUS (Version 7.0)* (Insightful, Seattle 2005)
51.50 R Development Core Team: *Writing R Extensions* (R Foundation for Statistical Computing, Vienna 2005)
51.51 J.M. Chambers: *Programming with Data: A Guide to the S Language* (Springer, Berlin, New York 1998)

52. Queuing Theory Applications to Communication Systems: Control of Traffic Flows and Load Balancing

With the tremendous increase in traffic on modern communication systems, such as the World Wide Web, it has made it imperative that users of these systems have some understanding not only of how they are fabricated but also how packets, which traverse the links, are scheduled to their hosts in an efficient and reliable manner. In this chapter, we investigate the role that modern queueing theory plays in achieving this aim. We also provide up-to-date and in-depth knowledge of how queueing techniques have been applied to areas such as prioritizing traffic flows, load balancing and congestion control on the modern internet system.

The Introduction gives a synopsis of the key topics of application covered in this chapter, i. e. congestion control using finite buffer queueing models, load balancing and how reliable transmission is achieved using various transmission control protocols.

In Sect. 52.1, we provide a brief review of the key concepts of queueing theory, including a discussion of the performance metrics, scheduling algorithms and traffic variables underlying simple queues. A discussion of the continuous-time Markov chain is also presented, linking it with the lack of memory property of the exponential random variable and with simple Markovian queues.

A class of queues, known as multiple-priority dual queues (MPDQ), is introduced and analyzed in Sect. 52.2. This type of queues consists of a dual queue and incorporates differentiated classes of customers in order to improve their quality of service. Firstly, MPDQs are simulated under different scenarios and their performance compared using a variety of performance metrics. Secondly, a full analysis of MPDQs is then given using continuous-time Markov chain. Finally, we show how the expected waiting times of different classes of customers are derived for a MPDQ.

Section 52.3 describes current approaches to assigning tasks to a distributed system. It highlights the limitations of many task-assignment policies, especially when task sizes have a heavy-tailed distribution. For these so called heavy-tailed workloads, several size-based load distribution policies are shown to perform much better than classical policies. Amongst these, the policies based on prioritizing traffic flows are shown to perform best of all.

Section 52.4 gives a detailed account of how the balance between maximizing throughput and congestion control is achieved in modern communication networks. This is mainly accomplished through the use of transmission control protocols and selective dropping of packets. It will be demonstrated that queueing theory is extensively applied in this area to model the phenomena of reliable transmission and congestion control.

The final section concludes with a brief discussion of further work in this area, an area which is growing at a rapid rate both in complexity and level of sophistication.

Queues, wherever they arise, are unfortunately an intrinsic part of human existence. We queue for items that are essential in our daily life as well as in situations that some would regard as an annoyance, although they are necessary, such as having to wait at a traffic intersection. On another level, modern communication systems, such as the internet, are under continuous strain in a world where it is not only demand for information that is increasing, but its speed of delivery. Given that some of this information has to be delivered over vast distances, is of varying sizes and is in competition for bandwidth with other traffic in the network, it has become essential in modern communication that traffic congestion be controlled, losses minimized and inefficient operations eradicated.

It has long been recognized that the problem of long delays suffered in many of our daily activities might be solved if one could model queues in all their manifestations. As a result, queueing theory was developed in the early part of the last century using tools and techniques from the well-established fields of probability and statistics to provide a systematic and general approach to understanding queueing phenomena. The earliest queueing applications are to problems of telephone congestion (pioneered by researchers such as *Erlang* [52.1] and *Palm* [52.2]). Subsequently, the subject was further developed and enriched with significant breakthroughs by researchers such as F. Pollaczek, A. Y. Khinchine, D. G. Kendall, D. R. Cox, J. R. Jackson, F. P. Kelly and many others. Queueing theory has been used to model many physical systems that involve delays, and currently, an important application is in modeling computer systems and communication networks.

This chapter considers the application of queueing theory to two critical issues of concern in modern communication systems, namely the problems of traffic flow control and load balancing, especially as they pertain to modern internet traffic. We begin in Sect. 52.1 with a brief review of basic queueing theory and then proceed to discuss the key topics of this paper in Sects. 52.2–52.4. The next three subsections provide a brief synopsis of these topics.

52.0.1 Congestion Control Using Finite-Buffer Queueing Models

Various scheduling algorithms have been introduced with the aim of improving quality of service (QoS) to customers. A wide variety of scheduling methods that aim to reduce congestion in communication systems have been studied. Many differentiate customers through marking and dropping processes (e.g. [52.3,4]). Others use time-marking and derivatives of this to allocate a degree of fairness in service, such as self-clocked fair queueing (SCFQ) and credit-based fair queueing (CBFQ) (e.g. [52.5,6]). A dual -queue length threshold (DQLT) [52.7] was used to divide real-time and non-real-time traffic to separate queues. Weighted round-robin (WRR) was looked at in [52.8].

In [52.9], a dual-queue (DQ) scheme was proposed to give better QoS to most customers at the expense of a few, rather than give poor QoS fairly to all customers. The dual-queue control scheme has two queues with finite space: namely the primary queue, which feeds into the service center, and the secondary queue, which acts as a waiting room when the primary queue is full (refer to Fig. 52.1). Upon arrival, a customer finding the primary

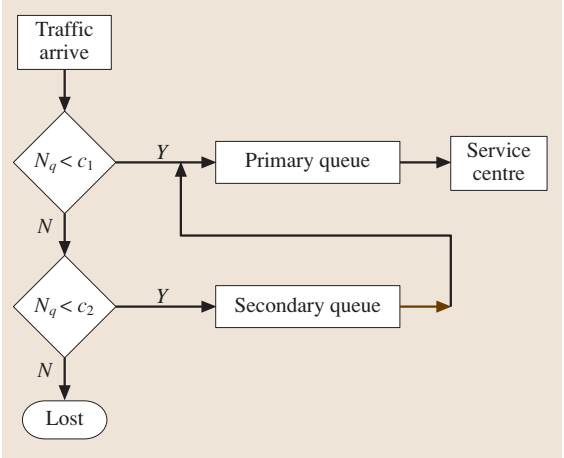


Fig. 52.1 The dual queue; N_q is the queue size, c_1 (c_2) is the primary (secondary) queue's buffer size

queue full waits in the secondary queue, if there is room. When a space becomes vacant in the primary queue, the customer at the front of the secondary queue joins the end of the primary queue. *Hayes* et al. [52.9] analyzed the delay characteristics of the dual queue against standard schemes such as first-come first-served (FCFS) and a modified deficit round-robin (DRR) technique [52.10], and demonstrated distinct advantages using the dual-queueing scheme. This work was extended to a wireless local area network [52.11] where minor modifications were made to the DQ and it was shown to outperform standard round-robin scheduling.

The multiple-priority dual queues (MPDQ), introduced in [52.12], builds upon this DQ scheme. The MPDQ introduces different classes into this scheme with the aim of providing better service to high-class customers without completely penalizing low-class ones. This is possible, not only because of the priority placed on customers' services, but also due to the MPDQ's partitioned queue structure. The MPDQ is especially relevant in the internet engineering task force (IETF) integrated services processes or differentiated services architecture. The MPDQ scheduling discipline provides a simple and effective mechanism for scheduling in these types of architecture.

52.0.2 Task Assignment Policy for Load Balancing

The usage of a *cluster* of commodity computers has become more prevalent in recent times. Such clusters are popular due to their scalable and cost-effective nature – often providing more computing resources at a significantly lower cost than traditional mainframes or supercomputers. They also provide other benefits, such as redundancy and increased reliability. The applications of such systems include supercomputing clusters and web-page serving for high-profile and high-volume websites, among other applications.

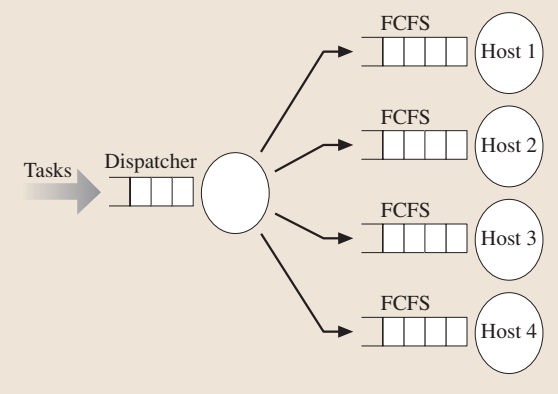


Fig. 52.2 Distributed server model

Figure 52.2 illustrates a common cluster configuration. Tasks, or *jobs* arrive at a central dispatcher, and are dispatched to hosts according to a *task assignment policy*. When a task arrives at the dispatcher, it is placed in a queue, waiting to be serviced in FCFS order.

The decision regarding which task assignment policy to utilize can significantly affect the perceived performance and server throughput. A poorly chosen policy could assign tasks to already overloaded servers, while leaving other servers idle, thus drastically reducing the performance of the distributed system. One major aim of a task assignment policy is to distribute tasks such that all available system resources are utilized and the load on the system is balanced. However, the ideal choice of task assignment policy is still an open question in many contexts.

The cluster configuration depicted in Fig. 52.2 is well suited to analysis via queuing theory. Equipped with some basic knowledge about our system of interest, such as the arrival and service distributions, we can easily obtain the expected performance metrics of the system. With these metrics, we can evaluate the performance of different task assignment policies, and make an informed judgment regarding which policy is best to employ.

52.0.3 Modeling TCP Traffic

The transmission control protocol (TCP) is a protocol that is widely used on the internet to provide reliable end-to-end connections. Reliability is achieved by verifying that each packet that enters the network is received correctly at the other end through the use of a return packet called an acknowledgment (ACK). It also provides congestion control, which attempts to prevent congestion collapse. Congestion collapse would occur if the amount of traffic entering the network greatly exceeded the capacity of the network. Congestion controls allows the amount of traffic entering the network to be controlled at the source.

The TCP congestion control mechanism uses a sliding window called a *congestion window* to control the rate at which packets are transmitted into the network. A congestion window has a size W measured in packets (actually its size is in bytes but it is simpler to think in terms of packets). This window is a segment of a larger buffer which starts at slot x and finishes at slot $x + W$. For example, a window of size $W = 3$ packets in a buffer

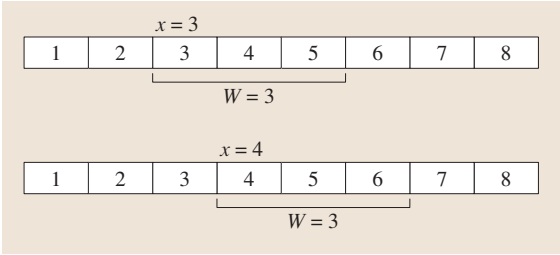


Fig. 52.3 Congestion window

of size eight slots could span slots either 1–3, 2–4, 3–5 and so on. Any packet within this window can be transmitted immediately into the network. When a packet is acknowledged by the receiving end, the window can slide across one place, allowing the next packet within the segment to be transmitted. For example if the window spans slot 3–5 then the window can slide to 4–6 and the packet can be transmitted in slot 6 when an ACK for packet 3 arrives (Fig. 52.3). This window size limits the maximum number of packets that can be in the network at any point in time.

The window size W changes over time as acknowledgements are received or lost. Queueing theory provides the ideal tool for analyzing the flow of packets within a network with TCP control, measuring its throughput and losses as packets are sent through a communication network.

52.1 Brief Review of Queueing Theory

The primary objective of queueing theory is to provide a systematic method of formulating, analyzing and predicting the behavior of queues. For example, customers waiting to be served at a store's checkout counter, cars waiting at an intersection controlled by traffic signals, telephone calls waiting to be answered and client request for an internet connection are a few of the countless phenomena that can be analyzed using standard queueing theory. The amount of literature devoted to queues and related problems is large and continues to grow at an exponential rate, especially since the advent of the World Wide Web. Published in 1961, the classic text on queueing theory, *Saaty* [52.13], has a list of over 900 papers in its bibliography. Since then, there have been many good texts on a broad range of queueing models, among which the following is a very short list of titles cited for their excellent coverage of key concepts and relevant examples: *Asmussen* [52.14], *Allen* [52.15], *Gross* and *Harris* [52.16], *Jaiswal* [52.17], *Kelly* [52.18] and *Kleinrock* [52.19, 20].

The basic queue is portrayed in Fig. 52.4. *Customers* arrive to the service facility from the environment and queue for service if there is someone ahead being serviced. After a length of time waiting, the customer is finally served, after which he departs from the queue. In this chapter, customers arrive as *single* units and not in *batches* or in *bulks*. Also, in the context of communication systems, the term *packets* will often be used interchangeably with *customers*. Each packet will be assumed to have a fixed uniform size; although in many systems packet sizes do vary, this does not affect the overall results unduly.

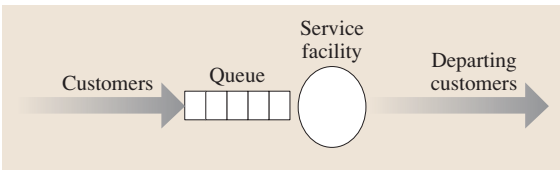


Fig. 52.4 The basic queue

The representation in Fig. 52.4 leaves out much of the internal working of the service facility, neither does it describes how customers arrive into the system. Mathematically, the process can be described more precisely in the following way. Customers arrive from an *input source* requiring service. The n-th customer C_n arrives at time T_n, $n = 1, 2, \ldots$, where $0 < T_1 < T_2 < \ldots$. The inter-arrival times $\tau_n = T_{n+1} - T_n$, $n = 1, 2, \ldots$ are usually assumed to form a *renewal process*, i. e. they are independent and identically distributed random variables. Customer C_n requires a service time of duration s_n and these service times for different customers are also independent and identically distributed random variables. In addition, the processes $(\tau_n)_{n\geq 1}$ and $(s_n)_{n\geq 1}$ are also assumed to be statistically independent of each other.

52.1.1 Queue Characteristics

The characteristics that determine a basic queues are

- A: arrival pattern of customers;
- B: service pattern of customers;
- C: the number of servers;

- D: system capacity or buffer size;
- E: service discipline.

The representation of a queue using the notation $(A/B/C/D)$ with D often omitted is due to David Kendall and is known as Kendall's notation.

Arrival Pattern

This is the distribution of the renewal sequence $(\tau_n)_{n\geq 1}$. Some common distributions are

$$\text{Exponential M:}$$
$$f(t) = \lambda\, e^{-\lambda t}\,,\ t \geq 0\,,\ \lambda > 0\,.$$
$$\text{Deterministic } D$$
$$\text{Erlangian with } k \text{ stages}$$
$$E_k : f_k(t) = \frac{\lambda(\lambda t)^{k-1}\, e^{-\lambda t}}{(k-1)!}\,,\ t \geq 0\,, k = 1, 2, \ldots\,.$$

If the exact distribution is unspecified, $A = GI$ for *general and independent*. Note that the symbol M in the case of the exponential distribution stands for Markov due to the fact that certain stochastic processes in queues, where either the inter-arrival distribution or service distribution are exponentially distributed, have the Markov property (cf. Sect. 52.1.4). Markovian queues are analytically more tractable than other general types of queues. We note also that arrivals to a queue could occur in bulks with the size of each bulk fixed or distributed as a random variable. Bulk and related queues are extensively covered in the book by *Chaudhry* and *Templeton* [52.21].

Service Pattern

This is the distribution of the sequence $(s_n)_{n\geq 1}$. In the internet system, the distribution is also called the *task size distribution* since it refers to the distribution of the sizes of files that are requested by random arrivals of requests to the internet. The most common distributions are the same ones given for the arrival pattern. However, for many communication systems such as the World Wide Web, so called heavy-tailed distributions are more appropriate for file sizes than the exponential distribution (cf. Sect. 52.2). Similar to the arrival pattern, service could also be implemented in bulk, i. e. instead of customers being served as single units, they are served in bulks of fixed or arbitrary sizes.

System Capacity

Traditional queueing theory assumes there is no limit on the number of customers allowed into the service facility. Obviously this is an approximation of what occurs in reality where there *is a limit* to the number allowed into *any* system. In communication networks for example, buffer sizes associated with links between service nodes are finite and packets arriving to a saturated link are *dropped*. This dropping of packets is very commonly employed in modern active queue management (cf. Sect. 52.4).

Number of Servers

The service facility is manned by one, or often by more than one, server who provide service to customers. In communication networks where there are no visible servers, the servers are replaced by the notion of *bandwidth*, which refers to the portion of a transmission link (in megabytes/second, MBps) that a class of packets is allocated. The larger the bandwidth allocated, the faster these packets will traverse the link.

Service Disciplines

Customers are selected for service by a variety of rules called *service disciplines* or *scheduling algorithms*. The most common service discipline is *first-come first-served* (FCFS) where the customer who arrives first is served first. However, many other types of scheduling algorithms are used in modern communication such as last-in first-out (LIFO), SIRO (service in random order) and prioritized service. A *priority queue discipline* is one where the servers specify certain rules for serving customers according to their classes [52.17]. This is usually implemented with *preemption* and *non-preemption*.

Preemption: A preemptive priority scheme is where a customer of a higher class interrupts and ejects a lower-classed customer from service. Three types of preemptive schemes are available: *preemptive–resume, preemptive repeat–identical* and *preemptive repeat–different*. The preemptive–resume scheme allows preempted customers to continue service from their initial point of interruption. The other two named schemes require customers to start again after preemption. The repeat–identical scheme allows the customer to begin again with the full amount of service time required, whereas the repeat–different scheme gives a random service time which does not take into account the time lost. Customers of the same class are served on a FCFS basis.

Non-preemption: The non-preemptive scheme is where a customer of a higher class must wait for a lower-classed customer to complete service if the latter was found to be in service upon the arrival of the higher-class customer.

One of the key objectives of this chapter is to highlight how different service disciplines have been used in communication systems and to compare their performance. We remark that the most mathematically

tractable system is the system $(M/M/c/K)$ with *FCFS* scheduling. Queueing systems with either $A = M$ and $B = GI$ or $A = GI$ and $B = M$ can be analyzed by the method of embedded Markov chains [52.22]. For more general systems, analytical results are often difficult or impossible to obtain. Such systems are conveniently solved using simulation techniques.

52.1.2 Performance Metrics and Traffic Variables

Most queueing models of communication systems assume that the prevailing conditions and constraints on the underlying processes are such that an *equilibrium steady state* is reached and one is dealing with a stationary situation. The analyses can then dispense with the time factor t. This situation will occur if the underlying process is *ergodic*, i. e. aperiodic, recurrent and non-null [52.19], and the system has been in operation for a long time.

The following is a glossary of some performance and traffic variables associated with the simple queue:

- λ : mean arrival rate of customers to the queue. This is usually assumed to be a constant, although it is generally a function of time t when the arrival process is nonstationary.
- μ : mean service rate of customers. This has the same qualification as for λ.
- ρ : traffic intensity. Defined by $\rho = \frac{\lambda}{p\mu}$ for a single class of customer arriving to a queue manned by p servers.
- L_q : the number of customers in the queue in the equilibrium state.
- L_s : the number of customers in the system in the equilibrium state, i. e. which also includes the customers being served.
- W_q : waiting time of each customer in the queue in the equilibrium state.
- W_s : waiting time of each customer in the system in the equilibrium state. In communication systems, W_s is also referred to as the *flow time*. Note that $E(W_s) = E(W_q) + E(X)$, where X is the service time of each customer and $E(\cdot)$ is the expected value operator.
- S : slowdown, defined by $S = W_q/X$. The expected slowdown, $E(S) = E(W_q)E(X^{-1})$ is an important metric in communication systems and measures the expected waiting time of each packet relative to its service requirement. This provides a fairer assessment of the delay suffered by a packet in a queueing system than the waiting time.

The following very useful results, known as Little's formulae, were established as a *folk theorem* for many years until they were shown to be valid by *Little* [52.23]:

$$E(L_q) = \lambda E(W_q)\,,$$
$$E(L_s) = \lambda E(W_s)\,. \tag{52.1}$$

For queues with finite capacity, λ in (52.1) is replaced by the *effective arrival rate* $\lambda_{\text{eff}} = \lambda \times [1 - \Pr(\text{queue full})]$.

52.1.3 The Poisson Process and the Exponential Distribution

Consider the arrival point process $(T_n)_{n\geq 1}$ and let $N(t)$ be defined as follows:

$$N(t) = \#\{n : T_n \in [0, t)\}$$

i. e. the number of arrivals that occur in the time interval $[0, t)$.

Definition 52.1

The point process $N(t)$ is a stationary Poisson process with rate $\lambda > 0$ if it has the following properties:

a) for any sequence of time points $0 = t_0 < t_1 < t_2 \ldots < t_n$, the process increments $N(t_1) - N(t_0)$, $N(t_2) - N(t_1)$, $\ldots$, $N(t_n) - N(t_0)$ are independent random variables;
b) for $s \geq 0$ and $t \geq 0$, the random variable $N(t+s) - N(t)$ has the Poisson distribution

$$P(N(t+s) - N(t) = j) = \frac{(\lambda s)^j \, e^{-\lambda s}}{j!}\,.$$

There is a close relationship between the exponential random variable and the Poisson process. Suppose the renewal sequence $(\tau_n)_{n\geq 1}$ has an exponential distribution with rate λ, i. e.

$$f(t) = \lambda e^{-\lambda t}\,,\ t \geq 0\,.$$

It can easily be shown that T_k has an Erlang distribution with k stages, i. e.

$$f_k(t) = \frac{\lambda(\lambda t)^{k-1} e^{-\lambda t}}{(k-1)!}\,,\ t \geq 0\,.$$

Therefore,

$$\Pr(T_k \leq t) = 1 - \sum_{j=0}^{k-1} \frac{e^{-\lambda t}(\lambda t)^j}{j!} \tag{52.2}$$

and, on using the obvious identity

$$\Pr(T_k \le t) = \Pr(N(t) \ge k) \tag{52.3}$$

we obtain

$$\begin{aligned}\Pr(N(t) = k) &= \Pr[N(t) \ge k] - \Pr[N(t) \ge k+1] \\ &= \Pr(T_k \le k) - \Pr(T_{k+1} \le t) \\ &= \frac{(\lambda t)^k \, \mathrm{e}^{-\lambda t}}{k!}\end{aligned}$$

after applying (52.2). Therefore a renewal sequence with an exponential distribution is a Poisson process. The converse also holds by applying (52.3) and assuming $N(t)$ is a Poisson process.

The Lack of Memory Property of the Exponential Random Variable

Amongst continuous random variables, the exponential random variable has the distinction of possessing the *lack of memory property* (for discrete random variables, the geometric random variable has that property). This property makes analysis of Markovian queues tractable.

The lack of memory property for X states that for any $t > 0$ and $s > 0$,

$$\Pr(X > t+s | X > t) = P(X > s) \, . \tag{52.4}$$

The exponential random variable clearly satisfies (52.4). Therefore, if the time X between consecutive occurrences of a Poisson process has already exceeded t, the chance that it will exceed $s+t$ does not depend on t. Thus the Poisson process *forgets* how long it has been waiting. From (52.4), we also have the following identity

$$\begin{aligned}\Pr(T > t + \Delta t | T > t) &= \mathrm{e}^{-\lambda \Delta t} \\ &= 1 - \lambda \Delta t + o(\Delta t) \, , \end{aligned} \tag{52.5}$$

which implies

$$\begin{aligned}\Pr(T \le t + \Delta t | T > t) &= 1 - \mathrm{e}^{-\lambda \Delta t} \\ &= \lambda \Delta t + o(\Delta t) \, , \end{aligned} \tag{52.6}$$

where $\Delta t \ll t$. Note that (52.5) is the probability of no events within the time interval $(t \, , t + \Delta t)$ given that the time from the last event to the next exceeded t, and (52.6) is the probability of an event within the time interval $(t \, , t + \Delta t)$ given that the time from the last event to the next exceeded t. Therefore, the probability of at least two events within $(t \, , t + \Delta t)$ is

$$1 - [1 - \lambda \Delta t + o(\Delta t)] - [\lambda \Delta t + o(\Delta t)] = o(\Delta t) \, , \tag{52.7}$$

i. e. in a small interval $(t \, , t + \Delta t)$, the Poisson process can increase by at most one occurrence with a non-negligible probability.

52.1.4 Continuous-Time Markov Chain (CTMC)

The theory of Markov process is fundamental in queueing theory and in other branches of applied probability. A comprehensive and authoritative account of the theory can be found in [52.24]. For our purpose, only some rudimentary knowledge of the theory is required and the results here will be presented without proofs.

Definition 52.2

A stochastic process X_t, $t \ge 0$, taking values in a discrete set S, which we may take as the set of non-negative integers, is a standard CTMC if, for any $n = 0, 1, \ldots$ and $t_0 < t_1 < t_2 < \ldots < t_n < t$ and values $i_0, i_1, \ldots, i_n$ and $j \in S$, the following identity holds:

$$\begin{aligned}&\Pr(X_t = j | X_{t_n} = i_n \, , X_{t_{n-1}} = i_{n-1} \, , \ldots \, , X_{t_1} \\ &\quad = i_1 \, , X_{t_0} = i_0) = \Pr(X_t = j | X_{t_n} = i_n) \, . \end{aligned} \tag{52.8}$$

A CTMC is said to be *stationary* if, for all $(i, j) \in S \times S$, the one-step transition probability $\Pr(X_{t+h} = j | X_h = i)$ is independent of h and we denote this probability by $p_{ij}(t)$. One-step transition probabilities for stationary CTMC satisfy:

1. $p_{ij}(t) \ge 0$ for all $(i, j) \in S \times S$ and $t \ge 0$.
2. $\sum_{j \in S} p_{ij}(t) = 1$ for all $i \in S$ and $t \ge 0$.
3. $$\lim_{t \to 0+} p_{ij}(t) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \ne j \end{cases} \, .$$
4. (Chapman–Kolmogorov equation)
$$p_{ij}(t+s) = \sum_{k \in S} p_{ik}(t) p_{kj}(s) \, .$$

Theorem 52.1

Let X_t, $t \ge 0$, be a stationary CTMC with transition probability functions $p_{ij}(t)$, $(i, j) \in S \times S$. Then the following (right) derivatives at $t = 0$ exist:

$$\lim_{t \to 0+} \frac{[p_{ii}(t) - 1]}{t} = -q_i$$

$$\text{and} \quad \lim_{t \to 0+} \frac{p_{ij}(t)}{t} = q_{ij} (i \ne j) \, .$$

The parameters q_{ij} and q_i are the *infinitesimal rates* of the CTMC. We note that $0 \le q_i \le \infty$. State i is an *absorbing* state if $q_i = 0$, and it is an *instantaneous* state if $q_i = \infty$. However, q_{ij}, $j \ne i$ is always finite.

Definition 52.3
The matrix $\boldsymbol{A} = (a_{ij})$ where

$$a_{ij} = \begin{cases} -q_i & \text{if } i = j \\ q_{ij} & \text{if } i \ne j \end{cases} \tag{52.9}$$

is called the *infinitesimal generator matrix* $\mathbf{A}$ of the CTMC X_t.

Definition 52.4
A CTMC process is *conservative* if

$$\sum_{j \ne i} q_{ij} = q_i < \infty \text{ for all } i \in S\,. \tag{52.10}$$

The sample path of a conservative CTMC can be described fairly succinctly. If the process is in state i, it remains there for a random time T_i which is exponentially distributed with cumulative distribution

$$\Pr(T_i \le t) = 1 - \mathrm{e}^{-q_i t}\,. \tag{52.11}$$

This is because, by the Markov property (52.8), the probability that the process next jumps to another state depends only on the last recorded state and not on how long it has been in that state. Hence, this lack of memory implies that T_i is an exponential random variable. Furthermore, given that the process is in state i, it next jumps to state $j \ne i$ with probability

$$P_{ij} = \frac{q_{ij}}{q_i}\,. \tag{52.12}$$

Therefore, by considering all situations that could occur, it follows that when $i \ne j$

$$p_{ij}(t) = \sum_{k \ne i} \frac{q_{ik}}{q_i} \int_0^t q_i\, \mathrm{e}^{-q_i s}\, p_{kj}(t-s)\,\mathrm{d}s \tag{52.13}$$

and when $i = j$

$$p_{ii}(t) = \mathrm{e}^{-q_i t} + \sum_{k \ne i} \frac{q_{ik}}{q_i} \int_0^t q_i\, \mathrm{e}^{-q_i s}\, p_{kj}(t-s)\,\mathrm{d}s\,. \tag{52.14}$$

Theorem 52.2
A conservative CTMC satisfies the following system of differential equations called the *forward equations*:

$$p'_{ij}(t) = \sum_{k \ne j} p_{ik}(t) q_{kj} - p_{ij}(t) q_j,\ (i, j) \in S \times S\,. \tag{52.15}$$

If we define the matrix $\boldsymbol{P}(t) = [p_{ij}(t)]$, (52.15) may be concisely expressed as

$$\boldsymbol{P}'(t) = \boldsymbol{P}(t)\boldsymbol{A}\,. \tag{52.16}$$

Let the state distribution of a CTMC X_t be denoted by $\pi(t) = [\pi_i(t)]$, where $\pi_i(t) = \Pr[X(t) = i]$, $i \in S$. Since

$$\pi(t) = \pi(0)\boldsymbol{P}(t)$$

we obtain from (52.16) the equation

$$\pi'(t) = \pi(t)\boldsymbol{A} \tag{52.17}$$

by pre-multiplying both sides of (52.16) with $\pi(0)$.

If the CTMC is ergodic, i. e. aperiodic, recurrent and non-null [52.19], the limit

$$\lim_{t \to \infty} \pi(t) = \bar{\pi}$$

exists and the rate of convergence is exponentially fast [52.24]; $\bar{\pi}$ is known as the *steady-state* or *equilibrium* distribution of the CTMC. Using (52.17), the steady-state distribution is obtained by solving the following system of equations called the *balance equations*:

$$\bar{\pi}\boldsymbol{A} = \boldsymbol{0}\,. \tag{52.18}$$

Birth–Death Processes

A *birth–death process* is a stationary conservative CTMC with state space S, the set of non-negative integers, for which the infinitesimal rates are given by

$$\begin{aligned} q_{n,n+1} &= \lambda_n\,, \\ q_{n,n-1} &= \mu_n\,, \\ q_n &= \lambda_n + \mu_n\,, \\ \text{and } q_{ij} &= 0\,,\ |i-j| \ge 2\,. \end{aligned}$$

A birth and death process can only move to adjacent states, with rates depending only on the state it has moved from (refer to Fig. 52.5). This CTMC is used to model queues where the renewal sequences $(\tau_n)_{n \ge 1}$ and $(s_n)_{n \ge 1}$ are exponential random variables. More

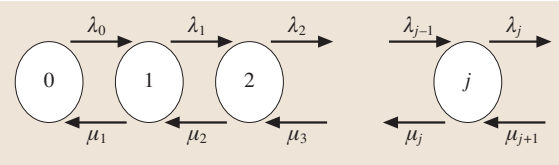


Fig. 52.5 Rate diagram of a birth–death process

specifically, the queueing processes $N_s(t)$ and $N_q(t)$ are birth–death CTMC since they can increase or decrease by 1 with an arrival to or departure from the queueing system, respectively.

From (52.17), the forward equations of the birth–death process take the form

$$\begin{aligned}\pi_0'(t) &= \mu_1\pi_1(t) - \lambda_0\pi_0(t)\,,\\ \pi_j'(t) &= \lambda_{j-1}\pi_{j-1}(t) - (\mu_j+\lambda_j)\pi_j(t)\\ &\quad + \mu_{j+1}\pi_{j+1}(t)\,,\ j\geq 1\,.\end{aligned} \tag{52.19}$$

The solution of (52.19) can be obtained through some ingenious arguments (see e.g. Chap. 4 in [52.13]) but it is much simpler to solve the corresponding balance equations (52.18) for the steady-state distribution $\bar{\pi} = (\pi_0\,,\pi_1\,,\ldots\,,\pi_j\,,\ldots)$, which are:

$$\begin{aligned}\lambda_0\pi_0 &= \mu_1\pi_1\,,\\ (\mu_j+\lambda_j)\pi_j &= \lambda_{j-1}\pi_{j-1} + \mu_{j+1}\pi_{j+1}\,,\ j\geq 1\,.\end{aligned} \tag{52.20}$$

If

$$\sum_{j=0}^{\infty}\frac{\lambda_j\lambda_{j-1}\ldots\lambda_0}{\mu_j\mu_{j-1}\ldots\mu_1} < \infty$$

then (52.20) admits the solution

$$\pi_j = \frac{\lambda_j\lambda_{j-1}\ldots\lambda_0}{\mu_j\mu_{j-1}\ldots\mu_1}\pi_0\,, \tag{52.21}$$

where

$$\pi_0 = \left(1+\sum_{j=0}^{\infty}\frac{\lambda_j\lambda_{j-1}\ldots\lambda_0}{\mu_j\mu_{j-1}\ldots\mu_1}\right)^{-1}\,. \tag{52.22}$$

In the next section, we apply the birth–death process to a very simple queueing model. This is done mainly to illustrate the calculations using the equations displayed in this section as well as to prepare the reader for an analogous but more involved analysis when we come to discuss the MPDQ.

Finite-Buffer Markov Queue (*M*/*M*/*c*/*K*)

This is a very old queueing model with application to telephony and is also commonly used to model other communication systems. Here, it is assumed that the facility can accommodate K customers, including the ones in service, and that once the service facility is full, new arrivals are not allowed in, i. e. they are *lost*. Arrivals to the system are generated according to a Poisson process with rate λ and service time is exponentially distributed with rate μ. There are c servers (or *lines*) where $c \leq K$. From the description of the model, it follows that

$$\lambda_j = \begin{cases}\lambda & \text{for } 0\leq j < K\\ 0 & \text{for } j\geq K\end{cases} \tag{52.23}$$

and

$$\mu_j = \begin{cases}j\mu & \text{for } 0\leq j < c\\ c\mu & \text{for } c\leq j\leq K\,.\end{cases} \tag{52.24}$$

Therefore, (52.21) and (52.22) give

$$\pi_j = \begin{cases}\pi_0\frac{(\rho c)^j}{j!} & \text{for } 0\leq j < c\\ \pi_0\frac{\rho^j c^c}{c!} & \text{for } c\leq j\leq K\,,\end{cases} \tag{52.25}$$

where $\rho = \frac{\lambda}{c\mu}$. Furthermore, applying (52.22), we obtain

$$\pi_0 = \begin{cases}\left(\sum_{j=0}^{c-1}\frac{(c\rho)^j}{j!} + \frac{(c\rho)^c(1-\rho^{K-c+1})}{c!(1-\rho)}\right)^{-1} & \text{if } \rho\neq c\\ \left(\sum_{j=0}^{c-1}\frac{(c\rho)^j}{j!} + \frac{(c\rho)^c(K-c+1)}{c!}\right)^{-1} & \text{if } \rho = c\,.\end{cases} \tag{52.26}$$

Note that π_K is the probability that a customer will be lost (or a packet *dropped*). The steady-state distribution can be used to obtain performance metrics such as:

$$E(L_s) = \sum_{j=1}^{K} j\pi_j$$

$$\text{and } E(L_q) = \sum_{j=c+1}^{K}(c-j)\pi_j$$

whence $E(W_s)$ and $E(W_q)$ are easily derived using the Little formulae (52.1) with λ_{eff} replacing λ.

52.2 Multiple-Priority Dual Queue (MPDQ)

The dual queue with finite buffer is illustrated in Fig. 52.1. We let c_1 be the capacity of the primary queue and c_2 be the capacity of the secondary queue. We assume arrivals belong to k differentiated classes of customers. All arrivals follow an independent Poisson process with rate λ_i and service times are independently exponentially distributed with rate μ_i for $i = 1, 2, \ldots, k$. Unlike the standard dual-queue analysis, we will be incorporating *class-based* service disciplines within the dual-queue model. This means that the favored (high) class of customers will jump to the head of the line before any other class of customers *within the same* queue. Therefore a lower-class customer in the primary queue cannot be ejected to the secondary queue even in the presence of higher-classed customers within the secondary queue. This differs from a single queue in that all high-class customers *in the system* are moved to the head of the queue, whereas in a dual queue, all the high-class customers *within each queue* are moved to the head of the line. A full primary queue blocks any entry from the secondary queue, hence there is an opportunity for lower-class customers to leave the system first, even in the presence of higher-class ones.

In the next section, we compare performances between single (SQ) and dual-queueing models based mainly on waiting-time analysis through simulations using the preemptive and non-preemptive service disciplines (preMPDQ and npMPDQ, respectively). We then give an overview of the analytical solutions when there are only two classes of customers in a subsequent section.

52.2.1 Simulating the MPDQ

The four scheduling disciplines used here in both the single- and dual-queue simulations are FCFS, LIFO, lowest class first (LCF) and highest class first (HCF). LCF and HCF are the two priority disciplines used in the MPDQ. In HCF (and the description is similar for LCF as the two disciplines are symmetrical), a higher-class customer jumps to the head of the queue, behind any already present customers of the same or higher class, and, in turn, is in front of any lower-class customer. As we will see from the simulation results, having both queues under the HCF regime in a dual queue is an effective form of traffic congestion control. The partitioning of a single queue and restricting entry into the primary queue, when it is full, allows lower-class customers to leave the system, even in the presence of other higher-class customers in the system. This is perceived as *fairness* to lower-class customers, as they can exit the system in the presence of higher-class customers but could be deemed as *unfair* to the latter. However, this can only occur when the primary queue is full, so higher-class customers are not disadvantaged all the time.

All simulations on the MPDQ were conducted using the Arena simulation software package [52.25], a flexible windows-based program suitable for a vast array of statistical analysis with dedicated queueing subroutines. In constructing the Arena model, two algorithms (for the preemptive and non-preemptive cases) were designed for the MPDQ. The single-queue schemes were also designed so that various regimes could be compared. Although our simulations dealt only with two classes of customers, which we label class 1 and class 2, all algorithms are simple to extend to multiple classes ($k \geq 3$) of customers. In communication systems, class 1 would refer to high-priority traffics such as web and voice traffic, and class 2 would refer to low-priority traffics such as file transfer protocol (FTP) and hypertext transfer protocol (HTTP).

Results of Simulation Experiments

All customers are considered to be of uniform length. For all models the first-class customer is considered the most infrequent arrival and longest in service. The ra-

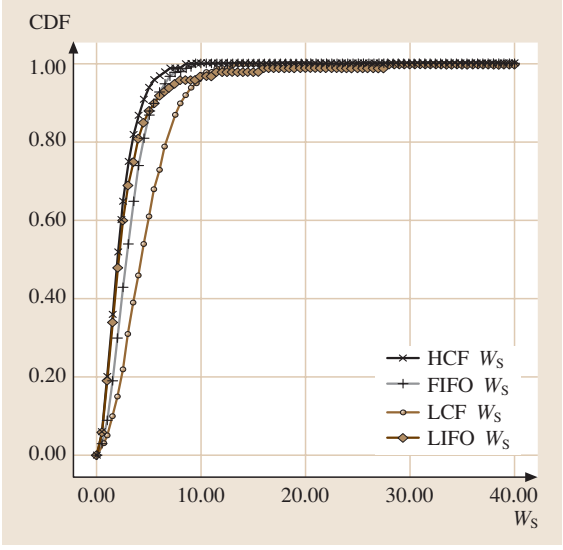


Fig. 52.6 Cumulative distribution functions (CDF) of waiting times for different scheduling disciplines

tionale is that first-class traffic will in many cases be the more demanding on system resources and can be seen as either the most or least valuable, depending upon the type of queueing discipline. Examples of high-class high-demand traffic include video-conference links, streaming audio or streaming video. As more classes are introduced, the performance differences between them may not be as apparent as when there are fewer classes.

The first analysis was to determine whether the HCF regime was superior to the LIFO, FCFS and LCF scheduling disciplines. Our first series of simulations involved the preemptive MPDQ model, the simplest of the two service regimes to simulate. We found that, as traffic intensity increased, the waiting time in the system was markedly lower under a HCF regime for class 1 customers. An example of this is seen in Fig. 52.6, for the parameters $\lambda_1 = 1, \lambda_2 = 2, \mu_1 = 1, \mu_2 = 2, c_1 = c_2 = 4$, the HCF has a lower probability of waiting in the system for class 1 customers in comparison to the other disciplines.

Loss probabilities were also examined. We found marginal differences between the regimes with respect to each class, and little difference in class-based loss. With no appreciable increase in loss by prioritizing traffic, and significant advantages in waiting time, the HCF discipline was chosen as the superior scheduling regime. Further simulations with non-preemptive models continue to confirm that HCF outperformed other scheduling regimes.

The next objective was to explore the differences between the non-preemptive and preemptive service disciplines. The results here provide a framework for communication providers in determining how service disciplines contribute to loss and delay, and how modeling, through simulation, can assist in reducing traffic congestion. We expect class 2 customers to be disadvantaged (in terms of expected waiting time) under the preemptive regime over class 1 (frequent ejection means longer waits) with the reverse situation under the non-preemptive regime. The results certainly show this to be true and Fig. 52.7 clearly demonstrates this.

What improvement does the npMPDQ offer over the non-preemptive single queue (SQ)? We compare the two by considering the non-preemptive dual queue with $c_1 = c_2 = 10$ and the SQ with capacity $c_1 + c_2 = 20$ with a view to measuring its performance under different traffic intensities, achieved through varying the mean arrival rate of class 2 customers. The graphs in Fig. 52.8 are based on a fixed class 1 mean arrival rate, and fixed mean service rates, being $\lambda_1 = 0.2$, and $\mu_1 = \mu_2 = 1$ respectively. We have varied the values of λ_2 from 0.25 to 20. This gives us scope to consider the performance of the system when the traffic intensity is low ($\rho = 0.45$) to when it is well beyond saturation.

From Fig. 52.8, we see that, in periods of low traffic intensity ($\lambda_2 < 1$), there is very little difference in the ratio of class 1 to class 2 customers in the npMPDQ and SQ models. However, for high traffic in-

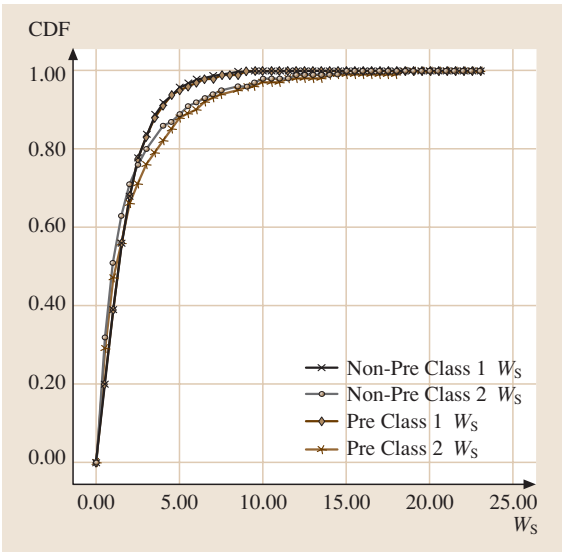


Fig. 52.7 Cumulative distribution functions (CDF) of waiting times for preemptive and non-preemptive service disciplines

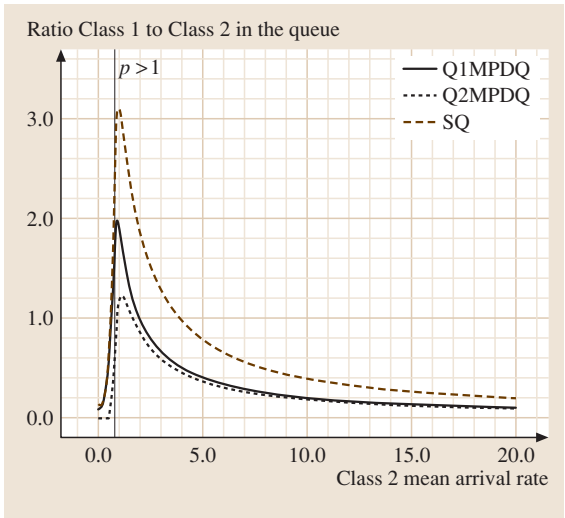


Fig. 52.8 Ratio of class 1 to class 2 customers

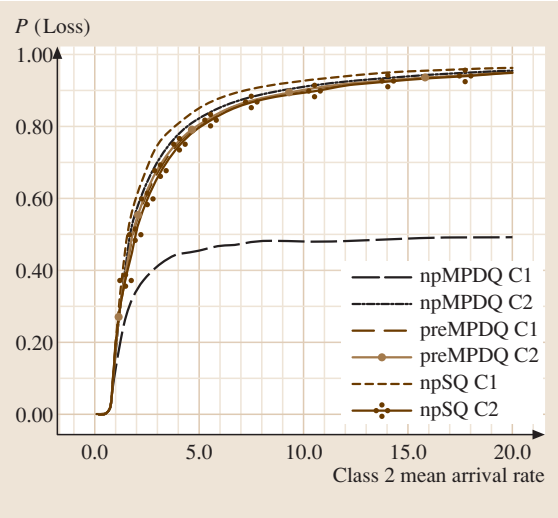


Fig. 52.9 Probability of loss

tensity ($\lambda_2 > 1$) and for both primary and secondary queues, the graphs of the MPDQ models peak and then decline much faster than the SQ model. Note also that, as traffic intensity increases, the distributions of the two classes in both queues appear to stabilize.

Finally we look at the probability of a loss. In Fig. 52.9, we note that class 1 non-preemptive MPDQ customers do best using this measure compared to all the others. The loss probability tapers off significantly for class 1 customers compared to both the SQ and preMPDQ. This does come at the expense of class 2 customers. However, the loss probability for class 2 is still superior to that of the class 1 SQ.

52.2.2 Solving the MPDQ Analytically

In this section, we give a brief summary of the work that has been undertaken in solving the MPDQ for its stationary distribution, and the derivation of the expected waiting times for both classes of customers ([52.12] and [52.26] respectively). We will assume the HCF discipline with class 1 customers designated as the high class. Furthermore, we will only consider the preMPDQ case here, the corresponding results for the npMPDQ case can be obtained from [52.27].

The State Space of the MPDQ and its Infinitesimal Generator

In order to solve the balance equations, which describe the movements by customers between and within the dual queues, it is necessary to define the state space S of the system and the infinitesimal generator matrix A containing the transition rates between states.

State space. The space S can be partitioned into two disjoint sets, $S = S_1 \cup S_2$, corresponding to the case when the secondary queue is empty and when it is not empty, respectively. Here,

$$S_1 = \{(i, j) : 0 \le i + j \le c_1\} ,$$

where i is the number of customers of class 1, and j is the number of customers of class 2. Similarly,

$$S_2 = \{(i, i', j') : 0 \le i' + j' \le c_2, i = 0, 1, \ldots , c_1\} ,$$

where i' is the number of customers of class 1 and j' is the number customers of class 2 in the secondary queue. Note that the number of class 2 customers in the primary queue is simply $c_1 - i$ in this case.

The states of the system can be labeled using S_1 and S_2 above according to the following lexicographical scheme:

$$\boldsymbol{i} = \{(i, 0) , (i, 1) , \ldots , (i, c_1 - i)\}$$
$$\boldsymbol{i0} = \{(i, 0, 1) (i, 0, 2) , \ldots , (i, 0, c_2)\}$$

and for $j = 1, 2, \ldots , c_2$

$$\boldsymbol{ij} = \{(i, j, 0) , (i, j, 1) , \ldots , (i, j, c_2 - j)\}$$

$i = 0, 1, 2, \ldots , c_1$.

The steady-state distribution vector $\overline{\pi}$ is thereby constructed so that its components are ordered using the above labeling scheme:

$$\overline{\pi}^t = \left(\overline{\pi}^t_{\mathbf{0}}, \overline{\pi}^t_{\mathbf{0,0}}, \ldots , \overline{\pi}^t_{\mathbf{0},\mathbf{c_2}}, \overline{\pi}^t_{\mathbf{1}}, \overline{\pi}^t_{\mathbf{1,0}}, \ldots , \overline{\pi}^t_{\mathbf{1},\mathbf{c_2}}, \ldots , \overline{\pi}^t_{\mathbf{c_1}}, \overline{\pi}^t_{\mathbf{c_1},\mathbf{0}}, \ldots , \overline{\pi}^t_{\mathbf{c_1},\mathbf{c_2}}\right) .$$

Note that the dimension of $\overline{\pi}$ equals $\frac{1}{2}(c_1 + 1)(c_2^2 + 3c_2 + c_1 + 2)$. The components of the above steady-state distribution can be described as follows: $\overline{\pi}_{\boldsymbol{i}}$ is the probability vector that is defined when there are no customers present in the secondary queue, whereas $\overline{\pi}_{\boldsymbol{ij}}$ is the probability vector that is defined when there are customers present in the secondary queue (so the primary queue is full). Any probability that is a component of the first type of vector has general form $\pi_{i,j}$ and any probability that is a component of the second type of vector has general form $\pi_{i,i',j'}$. For every tuple listed above, there is an additional label s representing the class of customer *in service* in case the MPDQ is non-preemptive. Thus, the dimension of the steady-state distribution vector is increased to $(c_1 + 1)(c_2^2 + 3c_2 + c_1 + 2) + 1$.

The infinitesimal generator matrix $\boldsymbol{A}$. Since the Markov process describing the MPDQ is an ergodic CTMC, the steady-state distribution $\overline{\pi}^t$ exists and is obtained by solving the system of balance equations

$$\overline{\pi}^t \boldsymbol{A} = \boldsymbol{0}\,, \tag{52.27}$$

where $\boldsymbol{0}$ is the zero vector and $\boldsymbol{A}$ is the infinitesimal generator matrix of the process. From the description of the dual-queueing system, $\boldsymbol{A}$ can be partitioned into submatrices whose detailed structure will be described in detail below. The general structure of $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{\Lambda}_0 & \boldsymbol{\Pi}_0 & 0 & \dots & \dots & 0 \\ \boldsymbol{\Omega}_1 & \boldsymbol{\Lambda}_1 & \boldsymbol{\Pi}_1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \boldsymbol{\Omega}_{c_1-1} & \boldsymbol{\Lambda}_{c_1-1} & \boldsymbol{\Pi}_{c_1-1} \\ 0 & \dots & & 0 & \boldsymbol{\Omega}_{c_1} & \boldsymbol{\Lambda}_{c_1} \end{pmatrix},$$

where

$$\boldsymbol{\Lambda}_0 = \begin{pmatrix} \boldsymbol{D}_0 & \boldsymbol{M}_{0,\lambda_2} & \boldsymbol{M}_{0,\lambda_1} & 0 & & 0 \\ \boldsymbol{O}_{0,\mu_2} & \boldsymbol{D}_{0,0} & \boldsymbol{Q}_{0,\lambda_1} & 0 & & \\ 0 & 0 & \boldsymbol{D}_{0,1} & \boldsymbol{R}_{0,1,\lambda_1} & & \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ & & & \ddots & \ddots & \boldsymbol{R}_{0,c_2-1,\lambda_1} \\ 0 & & \dots & & 0 & \boldsymbol{D}_{0,c_2} \end{pmatrix},$$

$$\boldsymbol{\Pi}_0 = \begin{pmatrix} \boldsymbol{N}_{0,\lambda_1} & 0 & & \dots & & 0 \\ 0 & 0 & & & & \\ \boldsymbol{S}_{0,\mu_2} & \boldsymbol{U}_{0,1} & \ddots & & & \vdots \\ 0 & 0 & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \dots & & 0 & \boldsymbol{U}_{0,c_2} & 0 \end{pmatrix},$$

and for $i = 1, 2, \dots, c_1$

$$\boldsymbol{\Lambda}_i = \begin{pmatrix} \boldsymbol{D}_i & \boldsymbol{M}_{i,\lambda_2} & \boldsymbol{M}_{i,\lambda_1} & 0 & & \dots & 0 \\ 0 & \boldsymbol{D}_{i,0} & \boldsymbol{Q}_{i,\lambda_1} & 0 & & & \\ \boldsymbol{S}_{i,\mu_1} & \boldsymbol{U}_{i,1} & \boldsymbol{D}_{i,1} & \boldsymbol{R}_{i,1,\lambda_1} & \ddots & & \vdots \\ 0 & 0 & \boldsymbol{U}_{i,2} & \boldsymbol{D}_{i,1} & \ddots & \ddots & \\ \vdots & & \ddots & \boldsymbol{U}_{i,2} & \ddots & \ddots & 0 \\ & & & \ddots & \ddots & \ddots & \boldsymbol{R}_{i,c_2-1,\lambda_1} \\ 0 & & \dots & & 0 & \boldsymbol{U}_{i,c_2} & \boldsymbol{D}_{i,c_2} \end{pmatrix},$$

$$\boldsymbol{\Omega}_i = \begin{pmatrix} \boldsymbol{P}_{i,\mu_1} & 0 & \dots & 0 \\ \boldsymbol{Y}_{i,\mu_1} & \boldsymbol{T}_{i,\mu_1} & & \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ & & & \\ 0 & & \dots & 0 \end{pmatrix},$$

and

$$\boldsymbol{\Pi}_i = \begin{pmatrix} \boldsymbol{N}_{i,\lambda_1} & 0 & \dots & 0 \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & 0 \end{pmatrix}.$$

Each submatrix represents all state transitions generated by the arrival/departure patterns of the MPDQ and incorporates the rate parameters λ_i and μ_i, $i = 1, 2$. We remark that not every submatrix is a square matrix and hence invertible. For example,

$$D_{i,j} = \begin{pmatrix} -(\mu_1+\lambda) & \lambda_2 & 0 & \dots & 0 \\ 0 & -(\mu_1+\lambda) & \ddots & & \\ & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \lambda_2 & 0 \\ & & \ddots & -(\mu_1+\lambda) & \lambda_2 \\ 0 & & \dots & 0 & -\mu_1 \end{pmatrix}$$

is a square matrix of dimension $(c_2 - j + 1)$ representing transitions between states in $\boldsymbol{ij}$, $i = 1, 2, \dots, c_1$, $j = 1, 2, \dots, c_2$ while

$$N_{i,\lambda_1} = \begin{pmatrix} \lambda_1 & 0 & \ddots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \vdots & & 0 & \lambda_1 \\ 0 & \dots & \dots & 0 \end{pmatrix}$$

is a matrix of dimension $(c_1 - i + 1) \times (c_1 - i)$ representing the transition from (i, j) to $(i+1, j)$ where $i = 0, 1, \dots, c_1 - 1$, $j = 0, 1, \dots, c_1 - i - 1$. Detailed of the other submatrices can be obtained from [52.12].

Technically speaking, any linear numerical procedure could be used to solve (52.27). However, this would ignore the structure of the system and the fine detail of the submatrices outlined above, especially the fact that

not all the submatrices are square matrices and that A is not in a workable block-tridiagonal form. This could render standard numerical procedures difficult or even impossible to apply; so, in [52.12], an algorithm which takes into account the structure of the system is proposed and shown to be very fast and easy to implement.

52.2.3 The Waiting-Time Distribution

The derivation of the waiting-time distribution, especially for class 2 customers, is based on the matrix analytic method pioneered by *Neuts* [52.28]. Our method generalizes the method proposed in [52.29] for the non-preemptive SQ. Matrix analytic methods exploit the special structures of the transition matrices or infinitesimal generators of some Markov processes occurring in queueing models and an important feature of these methods, which add to their utility, is that they are computational in character. In the subsequent analysis, we distinguish between two cases: when the primary queue is not full and when it is.

The Primary Queue is not Full

Class 1 customers: in this case, a tagged class 1 customer C_1 who joins the queue is concerned only with the number of class 1 customers ahead of him. If C_1 sees no class 1 customers ahead of him, then he goes to the head of the line and ejects the class 2 customer, if any, who is being served. If C_1 sees n, $0 < n < c_1$ class 1 customers ahead of him, then he has to wait until these customers have completed their services. Therefore, his waiting time is the sum of n exponential (μ_1) random variables. Let

$$\Pi_{Q_1} = \sum_{0 \le i+j < c_1} \pi_{i,j}$$

i. e. the probability that queue 1 is not full, then the conditional density of the waiting time of C_1 given that queue 1 is not full is

$$W_1^{(1)}(t) = \frac{1}{\Pi_{Q_1}} \left[\left(\sum_{j=0}^{c_1-1} \pi_{0,j} \right) \delta(t) + \sum_{j=0}^{c_1-2} \sum_{i=1}^{c_1-1-j} \pi_{i,j} f_{i,\mu_1}(t) \right] ,$$

where $\delta(t)$ is the Dirac delta function and

$$f_{n,\mu_1}(t) = \frac{\mu_1^n}{(n-1)!} t^{n-1} \mathrm{e}^{-\mu_1 t} ,$$

i. e. the Erlang distribution with n phases and parameter μ_1.

Class 2 customers: the derivation of the waiting time of an arbitrary class 2 customer C_2 entering the secondary queue is quite complex as it is affected by subsequent arrivals of both class 1 and class 2 customers. Suppose C_2 joins the queue and finds at least one customer in front of him, he will be pushed back in the queue by subsequent class 1 arrivals. Also, any subsequent class 2 arrivals will reduce the available spaces in the buffer, thus improving his chances of moving to the front of the line. The waiting time of C_2 will depend on the absorption time into a particular set $\mathcal{A}_1$ [defined by (52.28)] experienced by the stochastic process $Z(t)$, $t \ge 0$, where

$$Z(t) = [l_{11}(t), L_{21}(t), F_1(t)]$$

where $l_{11}(t)$ is the number of class 1 customers in the primary queue, $L_{21}(t)$ is the number of class 2 customers in front and including C_2 in the primary queue, and $F_1(t)$ is the amount of free space in the primary queue. Due to the assumptions underlying the MPDQ, $Z(t)$ is a continuous-time Markov chain taking values in the set

$$\mathcal{R}_1 = \{(l_{11}, L_{21}, F_1) \in \mathcal{N}_0^3 : 0 \le l_{11} \le c_1 , \; 0 \le L_{21} \le c_1 , \; 0 \le F_2 \le c_1 \} .$$

Define the set of states $\mathcal{A}_1 \subset \mathcal{R}_1$ by

$$\mathcal{A}_1 = \{(0, 1, F_1) : 0 \le F_1 \le c_1 - 1\} . \qquad (52.28)$$

Let $T_1(l_{11}, L_{21}, F_1)$ denote the first-passage time for the process $Z(t)$ into $\mathcal{A}_1$ starting initially in state (l_{11}, L_{21}, F_1) where $L_{21} \ge 1$, i. e.

$$T_1(l_{11}, L_{21}, F_1) = \min\{t \ge 0 : Z(t) \in \mathcal{A}_1 | Z(0) = (l_{11}, L_{21}, F_1)\} .$$

Note that this is precisely the time it will take for C_2 to be served. Denote the Laplace transform of $T_1(l_{11}, L_{21}, F_1)$ by $\hat{W}_{l_{11},L_{21},F_1}(s)$, i. e.

$$\hat{W}_{l_{11},L_{21},F_1}(s) = E(\mathrm{e}^{-sT_1(l_{11},L_{21},F_1)}) , \qquad (52.29)$$

where $\mathrm{Re}(s) > 0$. Using the Poisson arrivals see time averages (PASTA) property [52.30], which posits that a Poisson arrival would observe the steady-state distribution at any random time point, the Laplace transform of the time to absorption of $Z(t)$ given that C_2 can join the primary queue is therefore

$$\hat{W}_1^{(2)}(s) = \frac{1}{\Pi_{Q_1}} \left[\sum_{0 \le i+j < c_1} \pi_{i,j} \hat{W}_{i,j+1,c_1-i-j-1}(s) \right] . \qquad (52.30)$$

Inversion of (52.30) will in principle gives us the distribution of the waiting time of a class 2 customer. However, for most practical purposes, it suffice to compute its k-th moment, $k = 1, 2, \ldots$. A recursive and computationally efficient algorithm using matrix analytic methods is introduced in [52.26], which achieved this purpose.

The Primary Queue is Full

Class 1 customers: here, C_1 joining the secondary queue either sees no class 1 customers or at least one class 1 customer in the primary queue. In the first case, he has to wait for the services of all class 1 customers ahead of him in the secondary queue and that of the class 2 customer at the head of the queue to finish. In the second case, he has to wait until all class 1 customers in front of him in the combined queue have been served. Thus the conditional density of his waiting time given that the primary queue is full is

$$W_2^{(1)}(t) = \frac{1}{\Pi_{Q_2}} \left[\sum_{0 \le i'+j' < c_2} \pi_{0,i',j'} f_{1,\mu_2} \star f_{i',\mu_1}(t) + \sum_{i=1}^{c_1} \sum_{0 \le i'+j' < c_2} \pi_{i,i',j'} f_{i+i',\mu_1}(t) \right] \tag{52.31}$$

where $\star$ refers to the convolution operator and

$$\Pi_{Q_2} = \sum_{i=0}^{c_1} \sum_{0 \le i'+j' < c_2} \pi_{i,i',j'}$$

is the probability that the primary queue is full.

Class 2 customers: similar to the first case, the waiting time of an arbitrary class 2 customer C_2 is equal to the absorption time of a stochastic process into a targeted set. Define

$$Y(t) = [l_{11}(t), l_{12}(t), L_{22}(t), F_2(t)] ,$$

where $l_{11}(t)$ is the number of class 1 customers in the primary queue, $l_{12}(t)$ is the number of class 1 customers in the secondary queue, $L_{22}(t)$ is the number of class 2 customers in front and *including* C_2 in the dual queue, and $F_2(t)$ is the amount of free space in the secondary queue. The time for $Y(t)$ to first enter the targeted set $\mathcal{A}_2$ [defined by (52.32)] is equal to the waiting time of C_2. Note that $Y(t)$ is a continuous-time Markov chain which takes values in the set

$$\begin{aligned} \mathcal{R}_2 = \{(l_{11}, l_{12}, L_{22}, F_2) \in \mathcal{N}_0^4 : & 0 \le l_{11} \le c_1 , \\ & 0 \le l_{12} \le c_2 , \\ & 0 \le L_{22} \le c_1 + c_2 , \\ & 0 \le F_2(t) \le c_2 \} . \end{aligned}$$

The waiting time of C_2 is the first-passage time $T_2(l_{11}, l_{12}, L_{22}, F_2)$ for the process Y to enter into $\mathcal{A}_2 \subset \mathcal{R}_2$ defined by

$$\mathcal{A}_2 = \{(0, l_{12}, 1, F_2) : 0 \le l_{12} < c_2, 0 \le F_2 \le c_2\} \tag{52.32}$$

starting initially in state $(l_{11}, l_{12}, L_{22}, F_2)$, where $L_{22} \ge 1$. Denote the Laplace transform of $T_2(l_{11}, l_{12}, L_{22}, F_2)$ by $\hat{W}_{l_{11},l_{12},L_{22},F_2}(s)$. By the PASTA property, the Laplace transform of the time to absorption of Y into $\mathcal{A}_2$ given that a class 2 customer enters the secondary queue is

$$\hat{W}_2^{(2)}(s) = \frac{1}{\Pi_{Q_2}} \left[\sum_{i=0}^{c_1} \sum_{0 \le i'+j' < c_2} \pi_{i,i',j'} \times \hat{W}_{i,i',j'+c_1-i+1,c_2-i'-j'-1}(s) \right] . \tag{52.33}$$

Again, inverting (52.33) will technically gives us the distribution of the waiting time of a class 2 customer. However, as in previous case, a more viable alternative is to provide a recursive algorithm (which we have done in [52.26]) which will allow us to compute all the k-th, $k = 1, 2, \ldots$ moments of the waiting time for class 2 customers in the secondary queue.

52.3 Distributed Systems and Load Balancing

Many recent studies have shown that distributed computing environments exhibit a wide range of task sizes, often spanning many orders of magnitude. These so-called *heavy-tailed* workloads have been found to exist in a number of computing environments. *Crovella* et al. [52.31, 32] found that a number of file-size distributions measured on the World Wide Web (WWW) exhibit heavy tails, including file requests by users, files transmitted via the network and files stored on servers. Further examples of observed heavy-tailed workload include the size of files stored in Unix file systems [52.33] and the Unix process central processing unit (CPU) requirements measured at UC Berkley [52.34]. More recently, traffic measurements of the 1998 World

Cup [52.35] and the 1998 Winter Olympics [52.36] have exhibited some heavy-tailed characteristics. There are significant questions raised by these findings with regards to task assignment policies, as much of the existing work in the area was formulated under an assumption of an exponentially distributed workload.

Heavy-tailed distributions have very high variance, where 1% of tasks can take 50% of the computing resources. These distributions are characterized by the property

$$\Pr(X > x) \sim x^{-\alpha} ,$$

where $0 \leq \alpha \leq 2$. Any set of tasks that is said to follow a heavy-tailed distribution is described as having the following properties [52.37, 38]:

1. Decreasing failure rate. That is, the longer a task has run, the longer it is expected to continue running.
2. Infinite variance, and if $\alpha \leq 1$, infinite mean.
3. A very small fraction (less than 1%) of the very largest jobs make up a large fraction (half) of the workload. This is commonly referred to as the *heavy-tailed property*. It is this property that makes the load very difficult to balance effectively.

For the purpose of analysis, we assume that the task sizes show some maximum (but large) value. This is a reasonable assumption in many cases, such as a web server, which would have some largest file. A *bounded Pareto* distribution is therefore used, which has a lower and upper limit on the task size distribution. The probability density function for the bounded Pareto $B(k, p, \alpha)$ is:

$$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} x^{-\alpha-1}, k \leq x \leq p , \qquad (52.34)$$

where α represents the task size variation, k is the smallest possible task, and p is the largest possible task. By varying the value of α we can observe distributions that exhibit moderate variability ($\alpha \approx 2$) to high variability ($\alpha \approx 1$). Typical measured values of the α parameter are 0.9–1.3 [52.31, 32, 37], with an empirically measured mean of $\alpha \approx 1.1$. The next table provides some α values associated with the heavy-tailed distributions of some files.

52.3.1 Classical Load-Distribution Policies

The problem of optimal task assignment in a distributed system has been a well-researched area for many years. Most of the so-called *classical* approaches were created under the assumption that service times are exponentially distributed. Many of these policies are still widely used, due to their simplistic nature and ease of implementation.

Random and Round-Robin

Classical task assignment policies such as *random* and *round-robin* [52.39] have traditionally been used in distributed systems, and are still widely used for many applications. Under the random policy, tasks are assigned to each back-end server with equal probability. Using a round-robin policy, tasks are assigned to servers in a cyclical fashion. Both policies equalize the expected number of tasks allocated to each server, and are frequently used as a baseline to compare with other task distribution policies. Tasks are assigned with no consideration of each host's load or the distribution of task sizes. Despite this, random and round-robin are still commonly used in many scheduling environments (most likely due to ease of implementation). It has been shown previously [52.34] that random and round-robin both have similar performance characteristics.

Dynamic Policies

Dynamic policies intelligently assign tasks based on knowledge of the current load at each host. The LLF (least loaded first) approach assigns tasks to the server with the least amount of work remaining, attempting to achieve instantaneous load balance. The work remaining can be approximated by the queue length (shortest queue), or assuming the task's service requirement is known a priori. By keeping the load balanced, the waiting time in the queue can be reduced. It is known that balancing the load minimizes the mean response time [52.40] in the type of distributed system that we consider in this paper. Despite this, a number of caveats exist. Firstly, the best performance is not always obtained by balancing the load, particularly if you are interested in different measures of performance, such as the mean slowdown. Secondly, balancing the load is not always practical, as you are often depending on approximate measures of the load, such as the queue length. Under highly variable workload it is highly probable that the length of a queue can be misleading as an indicator of congestion.

Central Queue

The central-queue policy holds tasks in a queue at the dispatcher until a host is idle. Such a policy has proved to be equivalent to a least-work-remaining policy, showing that equivalent performance can be obtained without any prior knowledge of a task's size [52.37, 38]. Recently, two variations of the central-queue policy have

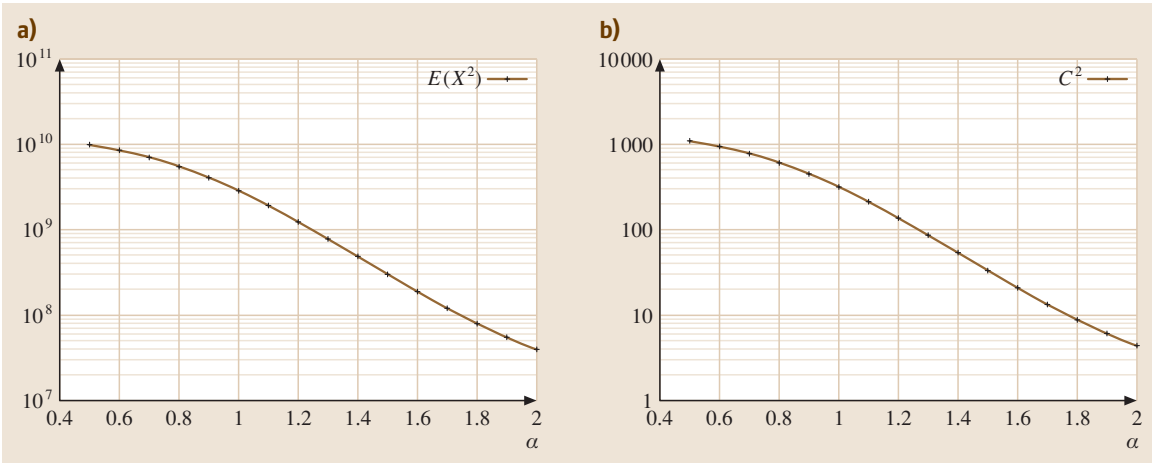


Fig. 52.10a,b The second moment of a bounded Pareto distribution ($E\{X\} = 3000$, $p = 10^7$) is shown in **(a)**, where α is varied from 0.5 to 2.0. The squared coefficient of variation ($C^2 = E\{X^2\}/E\{X\}^2$) is shown in **(b)**

been proposed – cycle stealing with immediate dispatch (CS-ID) and cycle stealing with central queue (CS-CQ) [52.41]. CS-ID immediately dispatches tasks to a back-end server, while CS-CQ holds tasks in a central queue at the dispatcher until a host is idle. Both policies are evaluated against a dedicated policy. In a dedicated policy, one host is *dedicated* to servicing all short jobs, while the other host services long jobs. Both CS-ID and CS-CQ follow a similar arrangement, but can steal cycles from an idle host if available and it is prudent to do so. Both CS-ID and CS-CQ show improvement over a dedicated policy in many areas (notably for short tasks). The application of these policies are limited to domains where a priori knowledge of a tasks size is known, and in the case of CS-CQ, there needs to be constant feedback between the dispatcher and the back-end hosts to notify the dispatcher of an idle host.

Some Known Results

The round-robin policy results in a slightly less variable arrival stream to the back-end hosts than the random policy. Despite this, the performance of the random and round-robin policies have been shown to be roughly equivalent [52.38].

Assuming an $M/M/c$ queueing system, a shortest-queue policy has been shown to be optimal with respect to maximizing the number of jobs completed by a certain time [52.42]. Under more general assumptions, *Nelson* and *Phillips* [52.43] claim that the central-queue (and therefore least-work-remaining) policy is optimal.

A least-work-remaining policy is not analytically tractable under $M/GI/c$ queuing systems. Nonetheless, this policy has been shown to be equivalent to a central-queue policy [52.38], for which there exists known approximations to its queue size using the following result given in [52.44]:

$$E(L_{M/GI/c}) = E(L_{M/M/c})\frac{E(X^2)}{E(X^2)}\,, \tag{52.35}$$

where X is the service requirement, and L_{model} represents the queue length under the model specified. We remark that, for the queueing model (M/GI/1), the expected waiting time in the queue is given by the famous *Pollaczek–Khinchin mean value formula* [52.19]

$$E(W_{\text{q}}) = \frac{\lambda E(X^2)}{2(1-\rho)}\,. \tag{52.36}$$

Limitation of Classical Load Distribution Policies

The task assignment policies listed above are not suitable for heavy-tailed distributions such as the Pareto distribution. Consider the metrics (52.35) and (52.36); what is immediately apparent is that all these metrics depend on the second moment of the service requirement distribution, $E\{X^2\}$ and for policies, such as the least-work-remaining policy, they will depend on the squared coefficient of variation $C^2 = E\{X^2\}/E\{X\}^2$. Figure 52.10 shows an example of one such highly variable distribution (where the y-axis is on a log scale). We can see that, as α decreases, the variation (as represented by the second moment of the distribution, and the squared coefficient of variation) increases substantially. Clearly,

as the variability of the service time distribution increases, the performance of the distributed system will decrease rapidly using these classical load balancing policies.

52.3.2 Size-Based Load Distribution Policies

In the previous section we have highlighted substantial recent research indicating the frequent occurrence of heavy-tailed workloads in many distributed computing environments. The characteristics of these heavy-tailed workloads make them very challenging to manage using traditional load distribution policies. Indeed, many of these policies were created under the $M/M/c$ queueing model, where the distribution of service requirements follows an exponential distribution. To deal with heavy-tailed $M/G/c$ workloads, where G is a heavy-tailed distribution, new load distribution techniques need to be employed. In particular, they must address the characteristics of these workloads, such as their highly variable nature, which cause such poor performance under traditional load distribution policies. In recent years, there have been several load distribution techniques specifically created to exploit the special characteristics of heavy-tailed workloads. They can be broadly classified as *size-based* policies where the workload is partitioned into distinct size ranges, with each size range dedicated to a specific server. For example, you may have a two server system where one server processes only small tasks, while another server processes only large tasks.

These size-based policies can be further classified by what knowledge they assume is known at the dispatcher. Some policies assumes that a task's size is known a priori at the dispatcher, and as such can assign the task directly to the server that is responsible for servicing tasks in that range. This obviously restricts the application of these policies to domains where exact (or reasonably accurate) a priori knowledge of a task's size is available.

Other size-based policies have less restrictive assumptions regarding what information is available at the dispatcher. Policies such as task assignment based on guessing size (TAGS) and task assignment based on prioritizing traffic flows (TAPTF), which are discussed later, assume no knowledge of a task's size at the dispatcher. They do, however, require knowledge of the distribution of task sizes.

SITA-E/V/U – Known Task Size

Size interval task assignment with equal load (SITA-E) [52.38] is a sized-based approach proposed by *Harchol-Balter* et al. that associates a unique size range with each host in the distributed system. These size ranges are chosen specifically to equalize the expected load received at each host. Whilst proving effective under conditions of high task-size variability, SITA-E is not the best policy in circumstances of lower task-size variability, where a dynamic policy is more suitable.

Size interval task assignment with variable load (SITA-V) [52.40] intentionally operates the hosts in a distributed system at different loads, and directs smaller tasks to lighter-loaded servers. The authors note that, depending on which performance metrics are of interest, the conventional notion that balancing the load on the respective hosts may not result in optimal performance, especially when the size distribution is heavy-tailed. SITA-V, like SITA-E, assigns tasks to a given host based on their size. However, SITA-V exploits the heavy-tailed property of the task size distribution by running the vast majority of tasks (i. e. the small tasks) on lightly-loaded hosts, while running the minority of tasks (the larger sized tasks) on the heavily-loaded hosts, thus preventing small tasks getting held up behind large tasks and allowing them to be processed quickly. Thus, mean slowdown is reduced, and the throughput is not adversely affected, but it can result in an increase in mean waiting time – which is expected, since minimal mean waiting time is known to occur when load is balanced.

A size-based approach that is specifically suited for batch computing environments under supercomputing workloads is size interval task assignment with unbalanced load (SITA-U) [52.45]. SITA-U purposely unbalances the load among the hosts while also being *fair*, i. e. achieving the same expected slowdown for all jobs. Two variations of SITA-U are considered: SITA-U-opt, where service requirement cutoffs are chosen to minimize mean slowdown, and SITA-U-fair, where service requirement cutoffs are chosen to maximize fairness. The simulation results showed that both variations of SITA-U performed better under a range of load conditions – with system loads in the range 0.1–0.8. SITA-U-fair achieved significant performance gains over the load range 0.5–0.8, demonstrating an improvement of 4–10 times with regards to mean slow down, and from 10–100 times with regards to variability in slowdown.

Most size-based policies perform well under very high task-size variation, but their advantage over existing approaches is reduced as variation decreases. Most importantly, the application of the task assignment poli-

cies listed above is limited by the assumption that the service requirement of each task is known a priori, which is frequently not the case.

TAGS/TAPTF – Unknown Task Size

The size-based approaches considered thus far all assume that the exact service requirements are known at the dispatcher in advance. Often this is not the case – in many environments a task's service requirement is not known until execution time on a given host. Task assignment based on guessing size (TAGS) [52.37] assumes no prior knowledge of a task's service requirement. Like SITA-V, TAGS is slightly counterintuitive in that it unbalance the load, and also considers the notion of *fairness*, i.e. that all tasks should experience the same expected slowdown. The TAGS approach works by associating a processing time limit with each host. Tasks are executed on a host, starting with host 1 (refer to Fig. 52.2), up until the designated time limit associated with that host; if the task has not completed by this point, it is aborted and restarted from scratch at the next host. These cutoffs are a function of the distribution of task sizes and the external arrival rate, and can be computed to optimize certain metrics, such as waiting time or slowdown.

The design of the TAGS policy purposely exploits properties of the heavy-tailed distribution, such as decreasing failure rate – where the longer a task has run, the longer it is expected to run – and the fact that a tiny fraction (less that 1%) of the very longest tasks can make up over half the load.

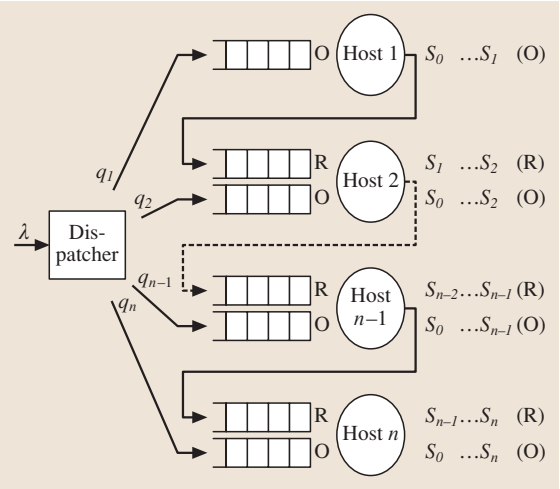


Fig. 52.11 TAPTF queueing system

Like other size-based approaches, under higher loads and less variable conditions, TAGS does not perform so well. TAGS gains much of its performance by exploiting the heavy-tailed property, by moving the majority of the load onto host 2, allowing the vast majority of small tasks to be processed quickly on host 1. TAGS also suffers under high loads due to *excess* – the extra work created by restarting many jobs from scratch. As noted in [52.37], " . . . overall excess increases with load because excess is proportional to λ, which is in turn proportional to load."

Recently a new approach to task assignment in a distributed system, task assignment based on prioritizing traffic flows (TAPTF) [52.46], was introduced (Fig. 52.11). TAPTF is a flexible policy that addresses some of the shortcomings of existing approaches to task assignment. TAPTF showed improved performance under heavy-tailed workloads for certain classes of traffic by controlling the influx of tasks to each host. Tasks can potentially be dispatched to any host, rather than just the first host, as in the TAGS approach. This approach is beneficial under two important scenarios. First, when task size variation decreases it becomes harder to exploit the so-called heavy-tailed properties. When the workload becomes more uniformly distributed, improved performance is gained from spreading the incoming tasks over multiple hosts. Second, when the system load is high the first host (which in a TAGS system receives *all* incoming tasks) can become overloaded, causing a bottleneck and resulting in a decrease in the overall performance metrics.

TAPTF introduces multiple queues with processing time limits (cutoffs). Each host has an ordinary queue which receives tasks directly from the dispatcher. Each host (excluding the first) also has a restart queue that receives restarted tasks from the host directly above it. The use of dual queues (combined with cutoffs at each host) enables service differentiation at each host, allowing smaller tasks to be executed quickly without being delayed by larger tasks. To achieve this, tasks that exceed the cutoff on a given host are migrated to the next host's restart queue (to be restarted from scratch). As the ordinary queue has priority over the restart queue, a potentially small task is not delayed by larger tasks that may exist in the restart queue.

Improved performances were observed both in mean waiting time and mean slowdown, the key areas where TAGS and random policies suffer. Most significantly, TAPTF exhibited improved performance under low to high task size variation and high system load by reducing

the excess associated with a large number of restarts and by intelligently controlling the influx of tasks to each back-end host.

Performance Under Heavy-Tailed Workloads

In this section, we illustrate how such size-based approaches are effective at counteracting the high variability in workloads that can affect the performance of traditional approaches to load balancing. Consider a size-based policy that partitions the workload between the back-end hosts. A size-based policy assigns unique size ranges to each of the n back-end hosts by partitioning the range of task sizes $[k, p]$, i. e. $k = s_0 < s_1 < s_2 < \ldots < s_n = p$. For example, in a two-host system, host 1 would service tasks with sizes between s_0 and s_1, while host 2 would handle the remaining tasks, with sizes between s_1 and $s_2 = p$.

Let p_i equal the fraction of tasks whose destination, i. e. where it will run to completion, is host i. That is, tasks whose size is between s_{i-1} and s_i. Using a bounded Pareto distribution (52.34), this is given by:

$$\begin{aligned} p_i &= P(s_{i-1} \le X \le s_i) \\ &= \frac{\alpha k^\alpha}{1-(k/p)^\alpha} \int_{s_{i-1}}^{s_i} x^{-\alpha-1}\,\mathrm{d}x \\ &= \frac{k^\alpha}{1-(k/p)^\alpha}\left(s_{i-1}^{-\alpha} - s_i^{-\alpha}\right) . \end{aligned} \tag{52.37}$$

Let us now consider only those tasks which are dispatched to and run-to-completion at host i. Let $E\left(X_i^j\right)$ be the j-th moment of the distribution of tasks that are dispatched to host i's queue. We have:

$$E\left(X_{iO}^j\right) = \int_{s_{i-1}}^{s_i} x^j f(x)\,\mathrm{d}x \tag{52.38}$$

$$= \begin{cases} \dfrac{\alpha s_{i-1}^\alpha}{(j-\alpha)\left[1-\left(\frac{s_{i-1}}{s_i}\right)^\alpha\right]} & \text{if } j \neq \alpha \\ \dfrac{s_{i-1}s_i}{s_i - s_{i-1}}(\ln s_i - \ln s_{i-1}) & \text{otherwise .} \end{cases} \tag{52.39}$$

Consider first the following example: a four-host system, utilizing the SITA-E task assignment policy. The arrival is Poisson and the service time distribution follows a bounded Pareto distribution. As described previously, SITA-E chooses its size ranges in order to equalize the expected load assigned to each back-end host. Figure 52.12a shows the C^2 values experienced by each back-end host as α varies. We can see a significant reduction in variation that has been achieved by partitioning the workload and assigning it to different hosts – effectively grouping like-sized tasks together. Indeed, the first two hosts have C^2 values that are less than one from $\alpha = 1.1$ to $\alpha = 2.0$. Nevertheless, we can see that the variation at the latter hosts is still quite high, approaching the value of C^2 of the task size distribution itself (i. e. before it is partitioned). This is not a problem in itself, as we can see in Fig. 52.12b. There, using the formula for p_i given in (52.37), we see that the vast majority of tasks are processed by the lower hosts, and predominantly the first host. These hosts have a significantly lower variance, and, given they process nearly

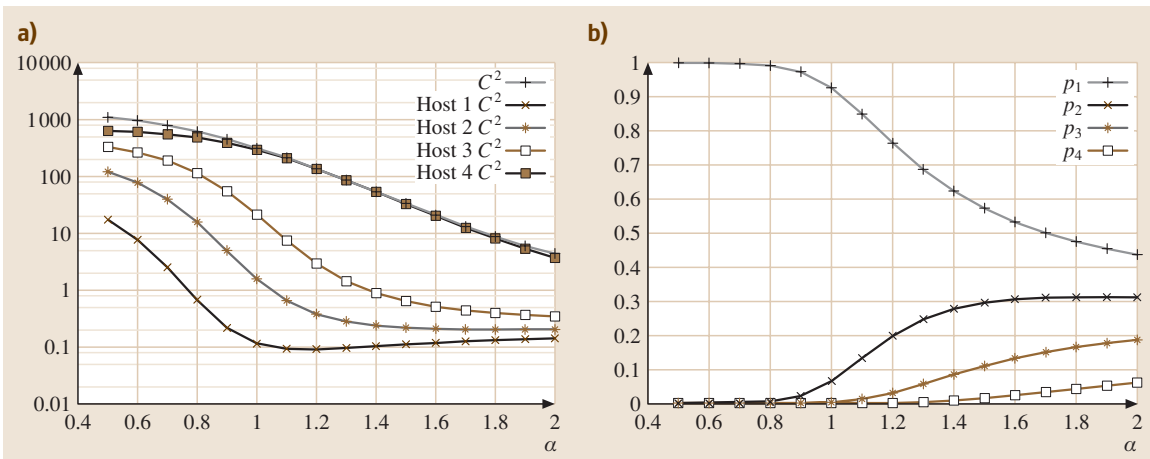


Fig. 52.12a,b The squared coefficient of variation experienced at each host ($C^2 = E\{X^2\}/E\{X\}^2$) in a four-host SITA-E system is shown in **(a)**. The fraction of tasks assigned to each host is shown in **(b)**

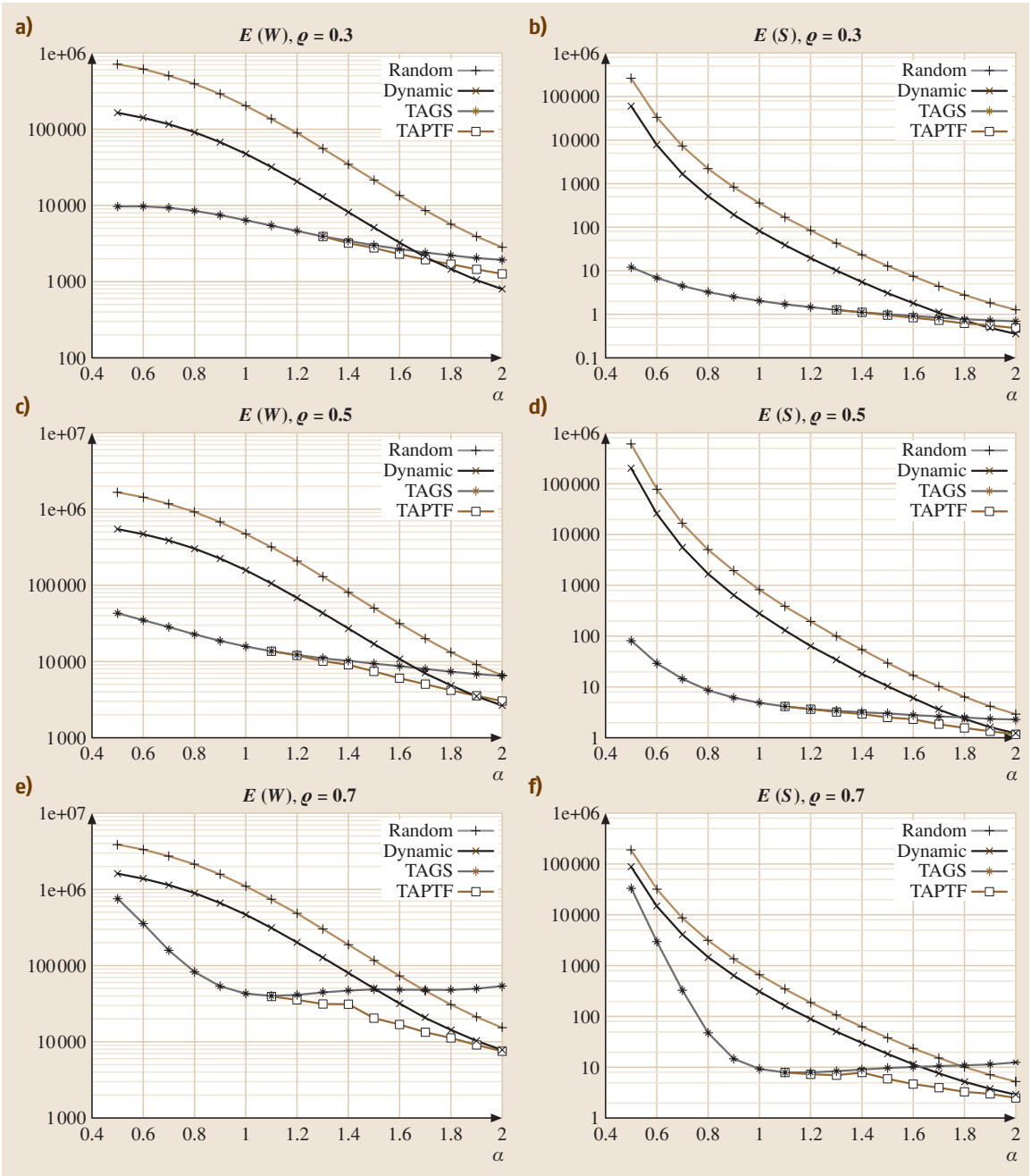


Fig. 52.13 Performance of a two-host distributed system with system load of 0.3, 0.5 and 0.7. The expected waiting time and slowdown are depicted under each load scenario

all tasks, they therefore contribute most to the overall system metrics, such as expected waiting time and slowdown. Recall that our system metrics depend on the second moment of the task size distribution; therefore, by reducing the variance experienced by the majority of tasks, size-based task assignment policies can signif-

icantly improve a system's performance over traditional techniques under heavy-tailed workloads. Next, we consider the performance of a distributed server cluster, where again our arrival is Poisson and our service time follows a bounded Pareto distribution. We set p (our largest task) to 10^7, and vary k (our smallest task) to keep the distributional mean, $E\{X\}$, fixed at 3000.

Figure 52.13 shows a comparison between the random, dynamic, TAGS and TAPTF policies in a two-host distributed system. The system load is varied, showing performance where $\rho = 0.3$ (low load), $\rho = 0.5$ (moderate load) and $\rho = 0.7$ (high load). The metrics used are the expected waiting time and mean slowdown.

As observed in Sect. 52.3.1, the performance metrics of both the random and dynamic policies are dependant on the variation of the task size distribution. As the variation increases (when α decreases) the expected waiting time and slowdown increase rapidly. Also, we note that the scale of improvement shown by the dynamic policy over the random policy decreases slightly as the system load increases. As the system load increases, there is a lower probability of a host being idle, even under the random policy. As such the dynamic policy has less scope for improvement.

Significantly, we can see an enormous improvement for the size-based task assignment policies, especially under conditions of high and extreme task size variations. These policies by their very nature reduce the variance of task sizes at each host, by partitioning the workload amongst each host. This has the effect of grouping similarly sized tasks together at the queues of each host, consequently reducing the variance at each host.

52.4 Active Queue Management for TCP Traffic

A description of how the TCP congestion control mechanism works, using a sliding congestion window to control the flow of packets through the system, is given in Sect. 52.0.3. The way in which the window size changes depends on the version of TCP used, the most common being the standard algorithms, slow start and congestion avoidance.

52.4.1 TCP Algorithms

Slow Start

Most versions of TCP that use the slow start algorithm are greedy and will attempt to take as large a share of the network resources as possible. The larger the window size a connection has, the larger the amount of resources it has. The purpose of slow start is to attempt to find the largest congestion window size a connection can have without causing too much congestion on the network. The following describes how slow start changes the window size from the commencement of a connection. In the algorithm described next, $W(a)$ refers to the window size after a acknowledgements.

1. The window size is set to one and a single packet is transmitted: $W(0) = 1$.
2. When an acknowledgement (ACK) for a packet arrives, the window size is increased by one and slides, allowing two new packets to be transmitted and acknowledging that the previous packet has been received correctly:
$$W(a+1) = W(a) + 1 \,.$$
3. When the window size reaches a slow-start threshold W_t (ssthresh), the slow-start phase ends and the congestion-avoidance phase begins.

Notice that the number of packets transmitted grows exponentially due to an increase in the number of ACKs received as the window size increases. For example, send one packet, receive one ACK, now send two packets and receive two ACKs, each of which can send a further two packets and so on.

Congestion Avoidance

Congestion avoidance usually commences after a slow-start packet has been lost; its purpose is to increase the window size slowly in an attempt to provide optimum utilization while preventing packet loss. The following are the steps of the congestion-avoidance algorithm:

1. After an ACK for one packet arrives, the window moves across one slot and one new packet can be transmitted.
2. After an ACK for every packet in an entire window arrives, the window size is increased by one, which allows two new packets to be transmitted. Therefore the window size changes according to (52.40):
$$W(a+1) = W(a) + \frac{1}{\lfloor W(a) \rfloor} \,, \qquad (52.40)$$

where $\lfloor x \rfloor$ refers to the *floor function*. For example, if $W(a) = 5$, then an arrival of an ACK would change the window to $W(a+1) = 5.2$ and five ACKs must arrive in total for the window to increase to 6.

3. Congestion avoidance ends when a maximum window size (MWS) is reached or when a packet loss occurs.

The above algorithm linearly increases the window size. The window size will only increase by one when all the ACKs from the window are received. Notice that to find the value of the window size at any a we must always take the floor of $W(a)$.

Retransmission Algorithms

In TCP packet loss is an indicator of congestion in the network and can occur in both slow start and congestion avoidance. Two common methods for detecting packet loss are fast retransmit and timeout.

Fast Retransmit. When the receiving end (client) acknowledges a packet it also indicates what the next required packet is. This will always be the packet that has the lowest sequence number. For example if the client received packets 1, 3 and 4 the next packet required is 2. Every time a new packet arrives packet 2 will be requested (in an ACK) until packet 2 arrives. These acknowledgements are called duplicate ACKs. Fast retransmit is based on the number of duplicate ACKs that are received which is usually three. If three duplicate ACKs arrive at the sender then fast retransmit will occur. This is only possible if at least three packets have been transmitted after the lost packet. Figure 52.14a shows the sequence of events in a fast retransmit where only one packet is lost. Notice packet 2 is transmitted after the third duplicate ACK for packet 2 arrives.

Timeout. Every time a packet is transmitted a timer is started, this timer will timeout, i. e. expire, after an estimated ACK arrival time. This estimated ACK arrival time is based on the mean and standard deviation of previous ACK arrival times. A timeout loss is detected when an ACK fails to arrive within the estimated arrival time and if three duplicate ACKs are *not* received. This means a timeout can only occur when a fast retransmit does not occur. Figure 52.14b shows the sequence of

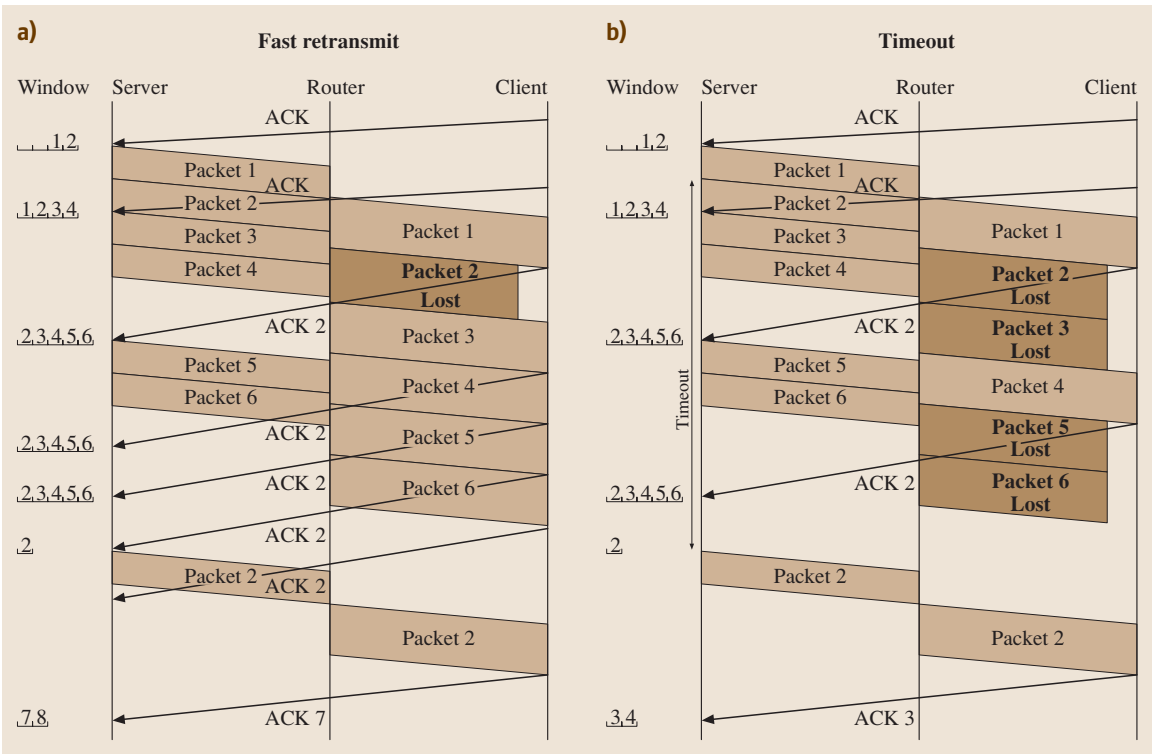


Fig. 52.14 Examples of fast retransmit and timeout

events for a timeout loss. Notice that only one packet is successfully transmitted after the first packet loss and only one duplicate ACK is received. Also notice that packets 5 and 6 are transmitted even though a packet loss has occurred.

52.4.2 Modeling Changes in TCP Window Sizes

Many models of TCP behavior have been proposed; most concentrate on finding the throughput of a TCP connections given some distribution of packet loss. Some models concentrate on short flows [52.47, 48] while other only consider long- or infinite-duration flows [52.49, 50]. A small number of models such as that of *Casetti* and *Meo's* [52.51] consider a mixture of long and short flows, which is the more realistic scenario. When short flows are modeled it is important to consider slow start because most short flows spend a majority of their time in slow start. For long flows, congestion avoidance is important because long flows spend most of their time in congestion avoidance. In [52.51], a model of the TCP congestion window size which represents both long and short finite-duration flows is presented.

In [52.52], the window size distribution is modeled as a Markov chain $\{U_k\}$, where k is the epoch of the change in window size. The state space of the process comprises four sets: active (N), idle (I), timeout (T) and fast retransmit (F), as shown in Fig. 52.15. N represents the set of states where data is available for TCP to transmit and members of this set are represented by the vector (W, W_t, N), where W is the window size, W_t the window threshold and W_M is the maximum received window size. The active states model the dynamic way the window size changes for both slow start and congestion avoidance. When a loss event occurs, the process makes a transition to one of two loss states, T and F, i.e to state (W, T) representing timeout has occurred, or (W, F) representing that fast retransmit has occurred. A transition to an idle set I models the situation when a connection has no data to send.

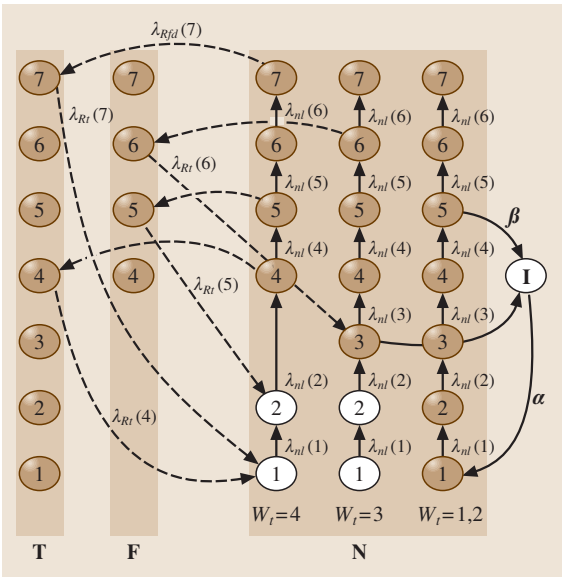


Fig. 52.15 Transitions between states in Markov model

From each of the active states N there are three different types of transition: timeout, fast retransmit and normal (i. e. no packets drop); each transition has a probability of occurrence of P_{Rt}, P_{Rf} and P_{nl}, respectively, which are used implicitly in calculating the transition rates between states. $U_k = (W, W_t, N)$ is an active transmitting state. All transitions from state U_k to U_{k+1} are summarized in (52.41) with their corresponding transition rates (the λs in the equations). For example if $W_M = 8$ and the current state is $(4, 2, N)$ then a transition to state $(5, 2, N)$ occurs if no packets are dropped. If a loss occurs then there is a transition to either state $(4, T)$, if it is a timeout, or $(4, F)$, if it is a fast retransmit. If there is no more data to send then there is a transition to state I.

$$U_{k+1} = \begin{cases} (2W, W_t, N)\lambda_{nl} & W < W_t \\ & 2W < W_t \\ (W_t, W_t, N)\lambda_{nl} & W < W_t \\ & 2W \geq W_t \\ (W+1, W_t, N)\lambda_{nl} & W_t \leq W < W_M \\ (W, T) & \lambda_{Rtd} \\ (W, F) & \lambda_{Rfd} \\ \mathrm{I} & \lambda_I \end{cases} \tag{52.41}$$

A timeout reduces the window size to one and the window threshold to $W/2$, which is half the window size, when a loss occurred. So from the timeout state $U_k = (W, T)$ a transition can occur to an active state $(1, \lceil W/2 \rceil, N)$ or the idle state if there is no more data to send, i. e.

$$U_{k+1} = \begin{cases} (1, \lceil W/2 \rceil, N) & \lambda_{Rt} \\ \mathrm{I} & \lambda_I\,. \end{cases} \tag{52.42}$$

Transitions from the fast-retransmit state $U_k = (W, F)$ are the same as the timeout state except the window size is reduced to half the loss window size instead of one, i. e.

$$U_{k+1} = \begin{cases} (\lceil W/2 \rceil, \lceil W/2 \rceil, N) & \lambda_{\mathrm{Rf}} \quad W > 3 \\ \mathrm{I} & \lambda_{\mathrm{I}} \,. \end{cases} \tag{52.43}$$

A transition from the idle state $U_k = \mathrm{I}$ to the initial active state $(1, \lceil W_M/2 \rceil, N)$ occurs with an arrival, i. e.

$$U_{k+1} = (1, \lceil W_{\mathrm{M}}/2 \rceil, N) \quad \alpha \tag{52.44}$$

where α is the arrival rate.

The stationary distribution π_{U} of $\{U_k\}$ can be obtained by solving the balance equations described by the transitions displayed in (52.41), (52.42), (52.43) and (52.44) in a similar way to that obtained for the MPDQ in Sect. 52.2.2. Note that the rates λ_{nl}, λ_{Rt} and λ_{Rf} are all functions of the window size W and must be defined from the packet-loss probability, loss distribution and delays of the network. A separate network model is required to define these parameters. Details of these derivations can be obtained from [52.52].

52.4.3 Modeling Queues of TCP Connections

The network that supports TCP connections is usually made up of a number of interconnected routers. Each router contains one or more buffers and a number of links. Bottlenecks can occur when a number of connections all attempt to utilize one link. Even though queueing can occur in any router, the bottleneck is usually most important because this is where the majority of queueing delay will occur. The most common type of queueing discipline assumed in routers are FCFS, the so called *droptail queues*. Recently active queue management (AQM) queues have been introduced specifically for TCP traffic. One of the most popular is the random early-detection queue (RED) [52.53]. In the following sections the FCFS droptail and RED queues will be described in more detail.

FCFS Droptail Queues

The most basic router queue on the internet is the droptail FCFS queue. A droptail queue has a finite buffer and drops packets if it is full. Packets arrive at the queue at the rate λ and are served at the rate μ. The arrival rate depends on the number of TCP connections connected to the queue and their maximum transmission rates. For example, if there are ten TCP connections with a maximum send rate of ten packets per second, then the maximum arrival rate would be one hundred packets per second. The departure rate μ is the number of packets that can be served on the outgoing link and depends on the bandwidth of the link. For example the outgoing link bandwidth may be 80 packets per second. The service time of each packet is deterministic because most packets of a TCP connection are the same size. The load of the network ρ is defined in Sect. 52.1.2. If the load is greater than one the network is considered congested and packets must be dropped from the queue. TCP aims to reduce the arrival rate of packets to the queue to a value close to one in an effort to minimize dropped packets and prevent packet retransmission, which adds further load.

An $M/D/1/K$ queue is a good model for approximating a queue of TCP connections because the arrival process of packets can be approximated as Poisson when there is a large number of sources, service time is deterministic and the queue has a finite capacity K. However, most of the operational parameters of an $M/D/1/K$ queue, such as average waiting time and drop probability P_K, i. e. the queue is full, cannot be obtained explicitly and the $M/M/1/K$ queue has been found to provide a good approximation [52.51]. As exhibited in Sect. 52.1.4, this simpler model has explicit equations for drop probability P_{K}, average waiting time $E(W_{\mathrm{q}})$ and average number of packets in the queue $E(L_{\mathrm{q}})$.

The average waiting time and packet drop probability together with the TCP send rate can be used to find the equilibrium point of a network using fixed-point analysis. The TCP send-rate equation finds the send rate (arrival rate to the queue) as a function of packet drop probability and average waiting time while the queue model finds the average waiting time and packet drop probability as a function of arrival rate.

Random Early Detection (RED)

As explained in previous sections, packet drops can control the send rate of a TCP connections. When the network contains droptail queues, the control of the TCP send rates is passive because only the queue length affects the packet drop probability. In active queue management the packet loss probability is controlled by the system administrator through various algorithms. These algorithms attempt to drop packets in strategic ways to control the send rate of TCP sources. One of the most common algorithms is RED [52.53].

RED attempts to control the rate of TCP traffic sources by dropping packets based on the average queue

Table 52.1 Some heavy-tail distributions

Description of files	α
Unix process CPU requirements [52.34]	1.0
Sizes of files transmitted over the internet [52.31,32]	1.1 – 1.3
1998 World Cup web site file size distribution [52.36]	1.37

size, estimated using exponential weighted moving average (EWMA). It attempts to keep the average queue size low and reduce delay. A number of thresholds are used by the RED algorithm, which are defined as follows:

- $\min_{\text{th}}$ – the average queue size at which packets will start to be dropped. When the average queue size is less than $\min_{\text{th}}$ no packets are dropped.
- $\max_{\text{th}}$ – the average queue size at which all packets will be dropped if it is exceeded.
- $p_{\max}$ – the maximum probability of dropping a packet.

Using these threshold, RED calculates and sets the probabilities p_b and p_a, which it uses in selecting packets to drop. These probabilities are given by (52.45) and (52.46), which are functions of average queue sizes avg:

$$P_b(\text{avg}) = \begin{cases} 0 & \text{avg} < \min_{\text{th}} \\ 1 & \text{avg} > \max_{\text{th}} \\ p_{\max}\frac{\text{avg}-\min_{\text{th}}}{\max_{\text{th}}-\min_{\text{th}}} & \min_{\text{th}} \le \text{avg} \le \max_{\text{th}} \end{cases} \tag{52.45}$$

$$P_a = \frac{P_b}{1 - i P_b}\,, \tag{52.46}$$

where i is the number of packets since the last dropped or marked packet. As the average queue size increases so does the probability that packets are dropped randomly from the queue. The relationship between the average queue size and drop probability p_b is linear within the $\min_{\text{th}}$ to $\max_{\text{th}}$ range (with values from 0 to $p_{\max}$, respectively). When the average queue size is below $\min_{\text{th}}$ no packets are dropped and when it is above $\max_{\text{th}}$ all packets are dropped. Using average queue sizes allows the queue to grow up to its maximum size in order to accommodate erratic (bursty) traffic flows. We refer the reader to [52.53] for details regarding the choice between using p_b and p_a in marking and dropping packets.

The RED algorithm is also useful in providing differentiated services because it can control the rate in which different classes of packets enter the network. In the next section we will investigate how RED is applied to provide differentiated services.

Table 52.2 Scheduling variables

Variable	Description
x	Priority state (high $x = 1$, low $x = 2$)
L	The queue that obtains service
Q_x	The instantaneous queue length of priority x queue
W_x	The weight assigned to the priority x queue
R_1	Average goodput, i. e. the amount of packets that gets transmitted through a system (packets/second)
T_1	Goodput threshold of high priority traffic

52.4.4 Differentiated Services

Differentiated services allow differing traffic to be categorized into a number of different service groups that provide different levels of service. For example, some packets may require low delay, while for others high throughput may be important. In a network, service is provided to a packet at the router buffers. The router can then schedule the order in which packets are allowed access to the link. The two common scheduling algorithms are weighted round-robin (WRR) and priority scheduling, which we have discussed earlier in Sect. 52.2. Another popular method of differentiating service at a buffer is by exploiting TCPs congestion control mechanism to control the rate at which packets enter the network. Packets can be actively dropped to signal to the TCP sources to slow down and prevent the packets from entering the network in the first place. Algorithms that achieved this are weighted RED (WRED) and RED in/out (RIO). These and other mechanisms of differentiated services will be explained in the following sections. For simplicity only two groups of service (high and low priority) will be assumed when describing differentiated services. Table 52.2 gives a list of variables that will be used in the ensuing discussions.

Weighted Round-Robin (WRR)

Weighted round-robin is used in class-based queues (CBQ) as an extension to round-robin scheduling (cf. Sect. 52.3.1) where each priority has a separate first-in first-out (FCFS) queue with a weight (refer to Fig. 52.16). Each queue is assigned a number of slots depending on its weight and for each slot, a queue can transmit a packet of data. For example, say there is a total of three slots and the high-priority queue is assigned two of the three slots, the low-priority queue has the remain-

ing slot. The slots are served in a round-robin fashion by a single server who moves from queue to queue so that the high-priority queue gets twice as much service as the low-priority queue. Note that packets are also serviced in order of priority with two-high priority packets serviced then followed by one low-priority packet, and so on. This can be formally described as follows:

$$S = W_1 + W_2 \,,$$

$$L = \begin{cases} 1 & Q_2 = 0,\, Q_1 > 0 \\ 1 & Q_1 > 0,\, (a \bmod S) \le W_1 \\ 2 & Q_1 = 0,\, Q_2 > 0 \\ 2 & Q_2 > 0,\, W_1 \le (a \bmod S) < S \end{cases} ,$$

where S is the total weight and the variable a is discrete and increments by one after each service or when a queue is empty.

The benefit of WRR is that no class denies any other class services, the low-priority class will always get its allocated amount of bandwidth. This provides a degree of fairness to all classes much akin to that provided by the dual queue in a MPDQ.

Priority Queues

A priority queue has been discussed in previous sections but in the context of customers rather than packets. Here, a priority queue can be a single queue which rearranges packets based on their priority (as is assumed in Sect. 52.2) or it can be multiple queues which have a priority of services. A single-priority queue involves complex rearrangement of packets within the queue, which can be processor-intensive and hence difficult to implement in practice. For this reason, we will mainly concentrate on a priority CBQ where each queue serves packets of a single priority and the service of the queues is prioritized, i. e. a queue with high-priority packets are always served before that with low-priority packets. The only time a low-priority packet can be served is when the high-priority queue is empty. The scheduling algorithm is then simply:

$$L = \begin{cases} 1 & Q_1 > 0 \\ 2 & Q_1 = 0,\, Q_2 > 0 \,. \end{cases}$$

As we have seen, in priority queues if the amount of high-priority traffic is large the low-priority traffic can be starved of service completely. The major advantage of priority queues in packet networks is that high-priority packets have much lower delay because they are always served first [52.54].

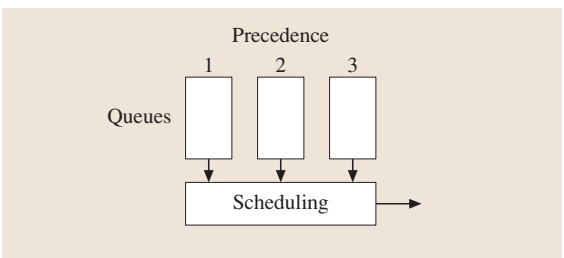


Fig. 52.16 A class-based queue

Flow-Based Quality of Service (QoS) with RED

A single flow is made up of many different types of packets, each of which can be given different levels of services. For example, a flow of size 60 packets could have 40 high-priority packets that must get to the other end and 20 low-priority packets that may get to the other end if there is available bandwidth. Weighted RED and RED in/out are a couple of ways that have been suggested to provide this differentiation within a flow.

Weighted RED (WRED). WRED [52.55] extends RED to allow different classes of packets to be treated differently. For example a high-priority class may have a lower probability of packet drop than a lower-priority class. WRED uses a single queue and maintains a single average queue size in exactly the same way as RED. It differs from RED by providing different $\min_{\text{th}}$, $\max_{\text{th}}$ and $p_{\max}$ for each class of packet. For example, WRED could be configured to have a larger $\min_{\text{th}}$ and $\max_{\text{th}}$ for high-priority traffic. This would make the high-priority packet-drop probability lower than that for low priority packets for all average queue sizes. A higher drop probability would mean that low-priority TCP sources would decrease their sending rate more than high-priority sources, thereby allowing more high-priority traffic through the link.

RED in/out (RIO). RIO [52.56] is similar to WRED except that it keeps a separate average queue length for the high-priority (*in*) packets and for the low-priority (*out*) packets. The reason different average queue sizes are used is to isolate the effect of *out* packets from *in* packets. For example if a single queue is used then a large load of *out* packets will increase the queue size and cause *in* packets as well as *out* packets to be dropped. By having a separate average queue size for *in* packets the number of *in* packets dropped is independent of the load of the *out* packets. RIO suffers the same prob-

lem as WRED in that when there is a large amount of *in* traffic it can starve low-priority *out* traffic of service.

Class-Based QoS with RED

A CBQ system can be combined with RED packet dropping to provide service differentiation between flows. Note this is a different level of service for an entire flow of packets rather than for individual packet. RED plus scheduling is an advantage over scheduling with FIFO because RED can control the source rate rather than just allocate desired bandwidth between different priority classes. For example if the high-priority traffic is allocated 1 Mbps out of 3 Mbps bandwidth, the high-priority load could be 2 Mbps. Rather than drop 1 Mbps of packets RED would attempt to adjust the load to 1 Mbps to fit the available bandwidth. The CBQ could use WRR or priority scheduling to serve each class of queue.

WRR RED. WRR RED simply combines WRR scheduling with RED. The WRR parameters are the same: each RED queue is allocated a number of service slots out of the total slots. Each RED queue can then use the same $\min_{\text{th}}$, $\max_{\text{th}}$ and $p_{\max}$, since the aim is to try to optimize the average queue length for the bandwidth allocated to the queue by WRR. WRR RED shares the advantages and disadvantages of WRR where high-priority packets are delayed by low-priority packets.

Table 52.3 DPRQ parameters

Parameter	Description
K	Maximum Buffer Size of both Queues
x	Priority (high $x = 1$, low $x = 2$)
D	Propagation delay
S_x	Number of priority x TCP sources
λ_x	Arrival rate of priority x packets
μ	Service rate of packets
P_{FS}	Probability of exceeding threshold given the current state is F and next state is S (above threshold S or $F = 1$, below threshold S or $F = 0$)
T_x	Goodput threshold of priority x queue
r_x	Average RED packet drop probability of priority x queue
g_x	Average RED packet no drop probability of priority x queue
d_x	Average total packet drop probability of priority x queue. RED plus queue full
q_x	Average queuing delay of priority x packets

Priority RED. Priority scheduling can also be combined with RED, just like WRR. Each RED queue is assigned a priority of services. For example if the highest-priority RED queue has packets in it, then it will always get service first. Priority RED faces the same advantages and disadvantages as the normal priority queueing discussed earlier. When there is a large amount of high-priority traffic it can starve low-priority traffic, with the advantage being that there would be a much lower delay for high-priority traffic.

Dynamic-Priority RED Queue (DPRQ)

Recently, we introduced the DPRQ [52.57], which extends the priority RED queue by adding a threshold to the amount of high priority traffic that is allowed service. The DPRQ scheduling algorithm allows a goodput i. e. successfully transmitted packets, threshold T_1 to be placed on high-priority traffic. The aim of the threshold is to prevent high-priority traffic from starving low-priority traffic, a drawback associated with the RED queues we have discussed earlier. The scheduling algorithm is as follows (Table 52.2):

$$L = \begin{cases} 1 & R_1 \le T_1,\ Q_1 > 0 \\ 2 & R_1 > T_1,\ Q_2 > 0 \\ 2 & Q_1 = 0,\ Q_2 > 0 \\ 1 & Q_1 > 0,\ Q_2 = 0 \end{cases} . \qquad (52.47)$$

When the average goodput of high-priority traffic R_1 is less than the threshold T_1 the high-priority queue will have priority in service. If R_1 increases beyond the threshold T_1 then it has exceeded its allowed goodput and the low-priority queue will be served with priority until R_1 is reduced. In this case R_1 naturally reduces because the high-priority queue is not being serviced. When there are no high-priority packets the low-priority queue will be serviced, if it is not empty. If the low-priority queue is empty then the high-priority queue is serviced. The average goodput R_1 is calculated over a specified time period using an exponential weighted average, just like that used to find the average queue length in the RED algorithm.

The DRPQ model and algorithm [52.57] will now be described in detail using the parameters which we have collected in Table 52.3.

As stated before in RED, the packet-drop probability depends on the average queue length. If the RED router is well configured the average queue length will not change a great deal over a large period of time. Over this period the average packet drop probability will be

Table 52.4 States of the DPRQ

Range	Description
$m = 0, n = 0$	Both queues empty
$m = 0, 0 < n < K$	High priority queue is empty
$m = 0, n = K$	High priority queue empty and the low priority queue full
$0 < m < K, n = 0$	Low priority queue empty
$m = K, n = 0$	Low priority queue empty and the high priority queue full
$0 < m < K, 0 < n < K$	General queue
$m = K, 0 < n < K$	High priority queue full
$0 < m < K, n = K$	Low priority queue full
$m = K, n = K$	Both queues full

P_b from (52.45). The probability of a packet not being dropped is therefore $g_x = 1 - P_b$, where x represents the priority of the queue.

Two possible states are defined for the threshold: either the threshold has been exceeded which is represented by a 1 or the threshold has not been exceeded, represented by a 0. The probability of the threshold changing from state F to state S is P_{FS}. For example, P_{01} is the probability, given that the threshold is currently not exceeded, that it will be exceeded in the next transition. Clearly, P_{FS} satisfy the following equations:

$$P_{00} + P_{01} = 1 ,$$
$$P_{10} + P_{11} = 1 .$$

We assume that the number of packets served in any time interval has a Poisson distribution, which is a valid assumption if the number of packets traversing the queue is large, as it is in this case. Therefore, the goodput R_1 is also Poisson with mean $\mu_1 = S_1 g_1 \lambda_1$ and standard deviation $\sigma_1 = \sqrt{S_1 g_1 \lambda_1}$. If the mean is large, which it is here since λ and S_x are usually large, the cumulative normal distribution can be used to approximate the cumulative Poisson distribution, i. e. we approximate R_1 by a normal random variable X. Therefore, for $y \geq 0$

$$P(R_1 \leq y) \approx P(X \leq y) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{y - \mu_1}{\sigma_1 \sqrt{2}}\right)\right] , \quad (52.48)$$

where $\operatorname{erf}(z) = \dfrac{2}{\sqrt{\pi}} \displaystyle\int_0^z \mathrm{e}^{-t^2}\,\mathrm{d}t$

is the error function. To approximate the probability that the threshold T_1 is not exceeded, simply set $y = T_1$ in (52.48) giving the following equation:

$$P_{00} = P_{10} = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{T_1 - \mu_1}{\sigma_1 \sqrt{2}}\right)\right] .$$

The queue length process is modeled using the Markov chain $\{V_k\}$ defined at the epoch of the k-th packet arrival to the queue. The states of $\{V_k\}$ are represented by the vectors (m, n, G), where m is the number of packets in the high-priority queue, n is the number of packets in the low-priority queue and G represents whether the threshold is exceeded in the high-priority queue: $G = 1$ signifies that the threshold is exceeded while $G = 0$ signifies that it is not. Table 52.4 indicates the possible range of states of V_k.

Different rate diagrams are needed for the different ranges shown in Table 52.4 but, due to lack of space, we will only show diagrams for the general range $0 < n < K$ and $0 < m < K$. We note that there are two sets of diagrams describing these transitions, one when $G = 0$ and another when $G = 1$. When the threshold has not been exceeded, the transitions from state $V_k = (m, n, 0)$ to V_{k+1} when $0 < m < K$ and $0 < n < K$ are described by (52.49) with corresponding transition rates:

$$V_{k+1} = \begin{cases} (m, n+1, 0) & g_2 \lambda_2 \\ (m+1, n, 0) & g_1 \lambda_1 \\ (m-1, n, 0) & P_{00}\mu \\ (m-1, n, 1) & P_{01}\mu . \end{cases} \quad (52.49)$$

When the threshold has been exceeded, the transition from $V_k = (m, n, 1)$ to V_{k+1} when $0 < m < K$ and $0 < n < K$ are described by (52.50).

$$V_{k+1} = \begin{cases} (m, n+1, 1) & g_2 \lambda_2 \\ (m+1, n, 1) & g_1 \lambda_1 \\ (m, n-1, 0) & P_{10}\mu \\ (m, n-1, 1) & P_{11}\mu. \end{cases} \quad (52.50)$$

Figure 52.17 shows the general transition-rate diagram. When the threshold is not exceeded the interactive

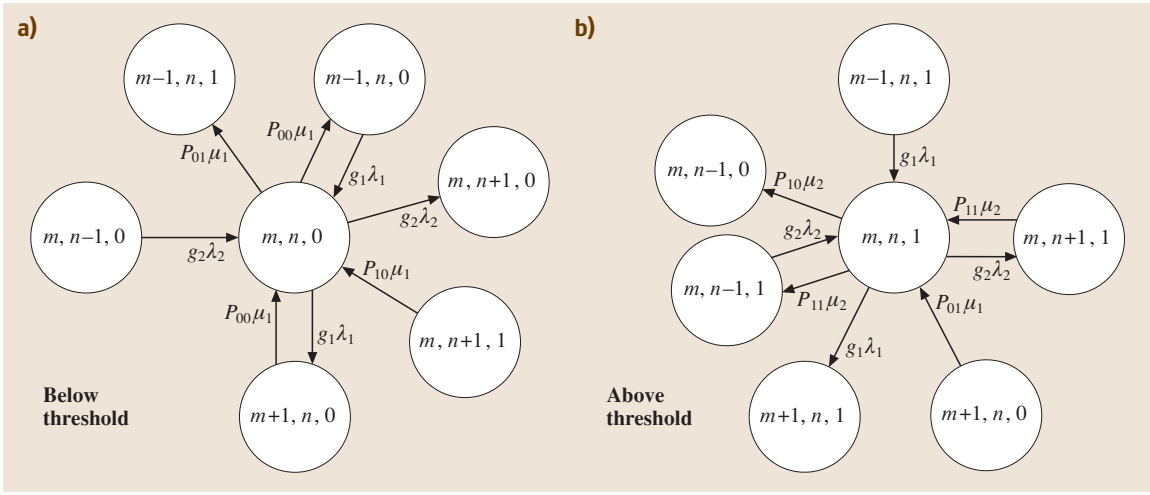


Fig. 52.17 DPRQ Markov process model ($0 < n < K$) and ($0 < m < K$)

queue has priority, therefore interactive packets are only served from states where the threshold has not been exceeded ($G = 0$ states). Similarly the low-priority queue is only serviced when the threshold has been exceeded therefore transitions can only occur from the low-priority queue when $G = 1$. Arrival of packets to either queue will cause the queue to increase in size.

Finally, extensive simulations based on the DPRQ and reported in [52.57] show general performance improvements, such as the reduction of packet loss, when compared to RED CBQ.

52.5 Conclusion

Modern communication systems, as exemplified by web server systems, continue to grow in complexity and level of sophistication. Queueing models have been extensively used to model and test new communication systems with the aim of improving QoS to users. In this chapter, we have discussed several approaches that have been advocated to reduce congestion and improve load balancing in these systems. We have considered both theoretical and practical aspects of analyzing and evaluating new systems, especially when the traffic flows are differentiated into different priority classes. The MPDQ discussed in Sect. 52.2 have been analyzed thoroughly, both in the theoretical and practical sense. However, a complete theoretical analysis of the network of multiple dual queues used in load balancing in Sect. 52.3 remains a challenge, as is the dynamic-priority RED queue (DPRQ) discussed in Sect. 52.4. All this is tied up with the feasibility of a comprehensive theoretical analysis of networks of queues with a specific type of node. Success in this venture would provide a unique opportunity to extend the range of problems that arise when designing new and reliable communication systems.

References

52.1 A. K. Erlang: Solution of some probability problems of significance for automatic telephone exchanges, Elektroleknikeren **13**, 5–13 (1917)

52.2 C. Palm: Intensitatsschwankungen im Fernsprechverkehr, Ericsson Technics **44**, 1–89 (1943)

52.3 M. A. Labrador, S. Banerjee: Enhancing application throughput by selective packet dropping, Proceedings of IEEE International Conference on Communications (ICC), Vancouver (1999) 1217–1222

52.4 W. Feng, D. D. Kandlur, D. Saha, K. G. Shin: Adaptive packet marking for maintaining end-to-end

throughput in a differentiated services internet, IEEE/ACM Trans. Network. **7**(5), 685–697 (1999)
52.5 S. G. Golestani: A self-clocked fair queueing scheme for broadband applications, Proceedings of the IEEE Infocom, Toronto (1994) 636–646
52.6 K. T. Chan, B. Bensaou, D. H. K. Tsang: Credit-based fair queueing (CBFQ), IEEE Electron. Lett. **33**(7), 584–585 (1997)
52.7 J. Jang, S. Shim, B. Shin: Analysis of DQLT scheduling for an ATM mMultiplexer, IEEE Commun. Lett. **1**(4), 175–177 (1997)
52.8 J. Wang, L. Yonatan: Managing Performance using Weighted Round Robin, IEEE Symposium on Computers and Communications **1**, 785–792 (2000)
52.9 D. Hayes, M. Rumsewicz, L. Andrew: Quality of service driven packet scheduling disciplines for real-time applications: looking beyond fairness, IEEE Infocom 1999 **1**, 405–412 (1999)
52.10 M. Shreedhar, G. Varghese: Efficient fair queuing using Deficit Round Robin, IEEE/ACM Trans. Network. **4**(3), 375–385 (1996)
52.11 R. Ranasinghe, L. Andrew, D. Hayes, D. Everitt: Scheduling disciplines for multimedia WLANs: Embedded round robin and wireless dual queue, Proc. IEEE Int. Conf. Commun. (ICC), ed. by K.-P. Estola, Helsinki (2001) 1243–1248
52.12 A. Bedford, P. Zeephongsekul: On a dual queueing system with preemptive priority service discipline, Eur. J. Oper. Res. **161**, 224–239 (2005)
52.13 T. Saaty: *Elements of Queueing Theory with Applications*. (Dover, New York 1961)
52.14 S. Asmussen: *Applied Probability and Queues* (Springer, Berlin Heidelberg New York 2003)
52.15 A. Allen: *Probability, Statistics and Queueing Theory: With Computer Science Applications* (Academic, New York 1990)
52.16 D. Gross, C. Harris: *Fundamentals of Queueing Theory* (Wiley, New York 1998)
52.17 N. Jaiswal: *Priority Queues* (Academic, New York 1968)
52.18 F. Kelly: *Reversibility and Stochastic Networks* (Wiley, New York 1979)
52.19 L. Kleinrock: *Queueing Systems 1: Theory* (Wiley, New York 1975)
52.20 L. Kleinrock: *Queueing Systems 2: Computer Applications* (Wiley, New York 1975)
52.21 M. L. Chaudhry, J. G. C. Templeton: *A First Course in Bulk Queues* (Wiley, Canada 1983)
52.22 D. Kendall: Stochastic processes occurring in the theory of queues and their analysis by the method of the embedded Markov chain, Ann. Math. Stat. **24**, 338–354 (1953)
52.23 J. D. C. Little: A simple proof of $L = \lambda W$, Oper. Res. **9**, 383–387 (1961)
52.24 K. L. Chung: *Markov Chains with Stationary Transition Probabilities* (Springer, Berlin Heidelberg New York 1960)
52.25 W. D. Kelton, R. P. Sadowski, D. A. Sadowski: *Simulation with Arena*, 2nd edn. (McGraw-Hill, New York 2002) p. 2
52.26 P. Zeephongsekul, A. Bedford: Waiting time analysis of the multiple dual queue with a preemptive priority service discipline, Eur. J. Oper. Res. (2006) (In press)
52.27 P. Zeephongsekul, A. Bedford: Analysis of the non-preemptive Multiple Priority Dual Queue, Department of Mathematics and Statistics Research Report **2** (2004)
52.28 M. Neuts: *Matrix-Geometric Solutions in Stochastic Models, An Algorithmic Approach* (John Hopkins Univ. Press, Baltimore 1981)
52.29 D. Wagner, U. Krieger: Analysis of a finite buffer with non-preemptive priority scheduling, Commun Stat.-Stoch. Models **15**, 345–365 (1999)
52.30 R. Wolff: Poisson arrivals see time averages, Oper. Res. **30**, 223–231 (1982)
52.31 M. Crovella, M. Taqqu, A. Bestavros: *Heavy-Tailed Probability Distributions in the World Wide Web* (Chapman Hall, New York 1998)
52.32 M. Crovella, A. Bestavros: Self-similarity in World Wide Web traffic: evidence and possible causes, IEEE/ACM Trans. Network. **5**(6), 835–846 (1997)
52.33 G. Irlam(1993): Available at http://www.base.com/gordoni/ufs93.html
52.34 M. Harchol-Balter, A. Downey: Exploiting process lifetime distributions for dynamic load balancing, ACM Trans. Comp. Sys. **15**(3), 253–285 (1997)
52.35 M. Arlitt, T. Jin: Workload characterization of the 1998 world cup web site, IEEE Network **14**, 30–37 (2000)
52.36 A. Iyengar, M. Squillante, L. Zhang: Analysis and characterization of large scale web server access patterns and performance, World Wide Web **2**, 85–100 (1999)
52.37 M. Harchol-Balter: Task assignment with unknown duration, J. ACM **49**(2), 260–288 (2002)
52.38 M. Harchol-Balter, M. Crovella, C. Murta: On choosing a task assignment policy for a distributed server system, J. Parallel Distrib. Comput. **59**(2), 204–228 (1999)
52.39 A. Silberschatz, P. Galvin: *Operating System Concepts*, 5th edn. (Addison-Wesley, Reading 1998)
52.40 M. Crovella, M. Harchol-Balter, C. Murta: Task assignment in a distributed system: improving performance by unbalancing load, Proceedings of ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Madison, USA 1998, ed. by S. Leutenegger (ACM Press, New York 1998) 268–269
52.41 M. Harchol-Balter, C. Li, T. Osogami, A. Scheller-Wolf, M. Squillante: Task assignment with cycle stealing under central queue, International Con-

ference on Distributed Computing Systems (ICDCS'03) **23**, 628–637 (2003)
52.42 R. W. Weber: On the optimal assignment of customers to parallel servers, J. Appl. Probab. **15**, 826–834 (1978)
52.43 R. D. Nelson, T. K. Philips: An approximation for the mean response time for shortest queue routing with general interarrival and service times, Perform. Eval. Rev. **7**(1), 181–189 (1978)
52.44 S. Sozaki, R. Ross: Approximations in finite capacity multiserver queues with Poisson arrivals, J. Appl. Probab. **13**, 826–834 (1978)
52.45 B. Schroder, M. Harchol-Balter: Evaluation of task assignmentpolicies for supercomputing servers. The case for load unbalancing and fairness, 9th IEEE Symposium on High Performance Distributed Computing , 211–220 (2000)
52.46 J. Broberg, Z. Tari, P. Zeephongsekul: Task assignment based on prioritising traffic flows, Lect. Notes Comp. Sci. **3544**, 415–430 (2005)
52.47 N. Cardwell, S. Savage, T. Anderson: Modelling TCP latency, Proc. IEEE Infocom, Isreal 2000, ed. by M. Sidi (IEEE, USA 2000) 1742–1751
52.48 T. Lakshman, U. Madhow: The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Trans. Network. **5**(3), 336–350 (1997)
52.49 A. Kumar: A comparative performance analysis of versions of TCP in a local network with a lossy link, IEEE/ACM Trans. Network. **6**(4), 485–498 (1998)
52.50 J. Padhye, D. Firouiu, D. Towsley, J. Kurose: Modeling TCP reno performance: a simple model and its empirical validation, IEEE/ACM Trans. Network. **8**(2), 133–145 (2000)
52.51 C. Casetti, M. Meo: A new approach to model the stationary behaviour of TCP connections, IEEE Infocom 1999 **1**, 367–375 (2000)
52.52 P. Dimopoulos, P. Zeephongsekul, Z. Tari: Modeling the burstiness of TCP, Proceedings of the IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecomunication Systems, Netherlands 2004, ed. by D. DeGroot, P. Harrison (IEEE, USA 2004) 175–183
52.53 S. Floyd, V. Jacobson: Random early detection gateways for congestion avoidance, IEEE/ACM Trans. Network. **1**(4), 397–413 (1993)
52.54 S. Floyd, V. Jacobson: Link-sharing and resource management models for packet networks, IEEE/ACM Trans. Network. **3**(4), 365–386 (1995)
52.55 Weighted random early detection, Tech. Rep. (CISCO Corp., USA 2003)
52.56 D. Clark, W. Fang: Explicit allocation of best-effort packet delivery service, IEEE/ACM Trans. Network. **6**(4), 362–373 (1998)
52.57 P. Dimopoulos, P. Zeephongsekul, Z. Tari: Reducing the user perceived delay of interactive TCP connections using a dynamic priority approach, Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN), USA 2005, ed. by E. K. Park (IEEE, USA 2005) 421–427

53. Support Vector Machines for Data Modeling with Software Engineering Applications

This chapter presents the basic principles of support vector machines (SVM) and their construction algorithms from an applications perspective. The chapter is organized into three parts. The first part consists of Sects. 53.2 and 53.3. In Sect. 53.2 we describe the data modeling issues in classification and prediction problems. In Sect. 53.3 we give an overview of a support vector machine (SVM) with an emphasis on its conceptual underpinnings. In the second part, consisting of Sects. 53.4–53.9, we present a detailed discussion of the support vector machine for constructing classification and prediction models. Sections 53.4 and 53.5 describe the basic ideas behind a SVM and are the key sections. Section 53.4 discusses the construction of optimal hyperplane for the simple case of linearly separable patterns and its relationship to the Vapnik–Chervonenkis dimension. A detailed example is used for illustration. The relatively more difficult case of nonseparable patterns is discussed in Sect. 53.5. The use of inner product kernels for nonlinear classifiers is described in Sect. 53.6 and is illustrated via an example. Nonlinear regression is described in Sect. 53.7. The issue of specifying SVM hyperparameters is addressed in Sect. 53.8, and a generic SVM construction flowchart is presented in Sect. 53.9. The third part details two case studies. In Sect. 53.10 we present the results of a detailed analysis of module-level NASA data for developing classification models. In Sect. 53.11, effort data from 75 projects is used to obtain nonlinear prediction models and analyze their performance. Section 53.12 presents some concluding remarks, current activities in support vector machines, and some guidelines for further reading.

53.1 Overview

The problem of predictive data modeling is of both academic and practical interest in many engineering and scientific disciplines, including software engineering. It is the process of building a model of the input–output relationship from historical or experimental data. This model is used to predict the output of a future occurrence for which only the input will be known. Such models have their roots in traditional statistics. However, recent advances in machine learning and related disciplines have been shifting focus away from statistical methods toward these approaches. In particular, a new type of learning machine, called a support vector machine (SVM), has gained prominence within the last decade. These machines are based on statistical learn-

ing theory, possess some very nice properties, and have exhibited impressive performance in a wide range of applications.

In this chapter we present the basic principles of support vector machines and their construction algorithms, with emphasis on applications. In Sect. 53.2 we formally describe data modeling for classification and prediction in software engineering and some important considerations in model development. Support vector machines are introduced in Sect. 53.3. Section 53.4 deals with the case of developing maximal margin classifiers for linearly separable classes and their relationship to the important concept of the Vapnik–Chervonenkis (VC) dimension. An illustrative example is used to explain the computations involved. Next, the more difficult problem of nonseparable patterns is presented in Sect. 53.5. Nonlinear classifiers using inner-product kernels are discussed in Sect. 53.6, and their computational steps are illustrated via an example. The development of nonlinear prediction models using the SVM algorithm is discussed in Sect. 53.7 and some comments about selecting SVM hyperparameters are summarized in Sect. 53.8. In Sect. 53.9, a generic SVM flow chart is presented to depict the development of classification and prediction models using SVM. In Sect. 53.10 a case study for module classification is detailed using public-domain software metrics data. Software effort-prediction using SVM nonlinear regression modeling is presented in Sect. 53.11 for some commercial software projects. A summary, a mention of current activities in SVM, and suggestions for further reading are included in Sect. 53.12.

For readers who want to get a general understanding of support vector machines, Sects. 53.2, 53.3, 53.4.1, 53.4.3, and 53.10 should be adequate. Section 53.6 is useful in understanding how SVM develops classifiers for practical classification problems. The remaining sections provide a description of related issues and SVM nonlinear regression.

53.2 Classification and Prediction in Software Engineering

53.2.1 Classification

A classification model is constructed from a set of data for which the attributes and the true classes are known and is employed to assign a class to a new object on the basis of its observed attributes or features. In software engineering, metric-based classification models are employed to classify a module as fault-prone or not fault-prone. Other terms that are used for module classes are high or low risk, high or low criticality, etc. An ability to identify fault-prone modules early in their lifecycle is of practical significance in software engineering because it enables allocation of appropriate resources such as additional reviews and testing to these modules and will thus enhance the quality of the delivered system.

Formally, the problem can be stated as follows. We have available metrics data ($\boldsymbol{x}_i$) about n modules and their corresponding class labels, y_i, denoted as

$$S = \left\{(\boldsymbol{x}_i, y_i), \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in (-1, +1)\right\}, i = 1, 2, \ldots, n, \quad (53.1)$$

where d is the dimensionality of the input metrics data. Here the value of y_i as -1 or $+1$ is based on some threshold value of the numbers of faults found in the module. The modeling task is to construct a support vector classifier (SVC) that captures the unknown pattern between the inputs and the outputs, based on the evidence provided by the data. The developed model is then used to predict the criticality class of a future module whose software metrics values ($\boldsymbol{x}$) would be known, but not the class (y).

The objective is to develop a model with good accuracy on both the training data and on future predictions. A common measure used for assessing accuracy is the classification error (CE), which is the ratio of the number of modules classified correctly to the total number of modules. To seek an objective measure of the future performance of the classification model, one approach is to partition the sample data randomly into three sets: training, validation, and test, as shown in Fig. 53.1. The first set is used for model development, i. e., for training.

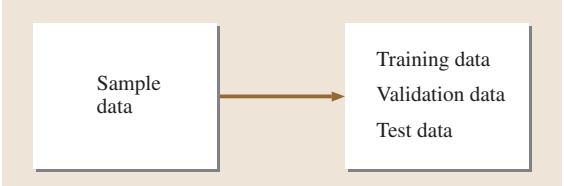


Fig. 53.1 Sample-data partitioning

The candidate models are evaluated on the validation data and usually the model with the smallest CE is selected. Then, the selected model is used to classify the data in the test set, and this error is considered to be an objective measure of future performance, or generalization error, of the selected model. The relative sizes of these sets are application-dependent. However, 50% of the data is often used for training, 25% for validation, and 25% for testing. This is called the *hold out* approach. Sometimes, only two subsets are created, training and the test, and the test set is used for model selection.

If the data set is small, the generalization error is estimated using cross validation. For k-fold cross validation (KCV), the data set is randomly divided into almost equal k sets and the model is developed from $(k-1)$ sets and tested on the omitted k-th set. The process is repeated k times, leaving a different set out each time. The average CE on the k omitted sets is called the *KCV error* of the model. Commonly used values for k are five and ten. Sometimes, we use $k = 1$. Then, the estimated error is called the *leave one out* (*Loo*) error.

53.2.2 Prediction

The development of such models can also be seen as finding input–output mapping, but now between the project features and the effort. This problem can be stated as follows. Given some input and output data about previous projects, find a suitable functional relationship between them. Suppose we are given a training data set,

$$S = \left\{ (\boldsymbol{x}_i, y_i), \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R} \right\}, i = 1, 2, \ldots, n \,, \tag{53.2}$$

where $\boldsymbol{x}$ are the software project features, d is the number of features, y is the effort, and n is the number of projects in the data set. Then, the goal is to find a function or mapping that maps $\boldsymbol{x}_i$ to y_i, $i = 1, \ldots, n$, such that the training and generalization errors are small. Two error evaluation measures commonly employed in software effort research are the mean magnitude of relative error (MMRE) and PRED(25), defined as follows:

$$\text{MMRE} = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| \text{Actual effort}, y_i - \text{Predicted effort}, \hat{y}_i \right|}{\text{Actual effort}, y_i} \,; \tag{53.3}$$

$$\text{PRED}\,(25) = \frac{\text{Number of estimates within 25\% of actual } y}{n} \,. \tag{53.4}$$

These are measured for the model using the training data set and evaluated on the test data, usually by the Loo cross validation, since the number of projects used for effort estimation is generally small. Note that MMRE is a measure of error, while PRED(25) is a measure of accuracy. Therefore, we seek low MMRE and high PRED(25) values. The model performance on test data is used as a measure of its generalization ability, that is, a measure of the predictive accuracy on future projects for which the effort would be estimated.

53.3 Support Vector Machines

In this section we provide a very brief introduction to support vector machines. Consider the classification and regression tasks we just discussed. The statistical techniques for dealing with these require a knowledge of the underlying distribution. In many practical situations this distribution is not known and has to be assumed. Due to this restriction we seek alternative, distribution-free procedures. A support vector machine provides such a procedure that is relatively new and has some very nice properties. In particular, SVM is a learning procedure that can be used to derive learning machines such as polynomial approximators, neural networks, and radial basis functions. The origins of SVMs are in statistical learning theory (SLT) and they represent an approximate implementation of the principle of structural risk minimization (SRM). These are discussed in-depth in [53.1, 2] and are beyond the scope of this chapter. Briefly, the SRM principle is based on the fact that the test error of a learning machine, that is, the generalization error, is bounded by the sum of two quantities. The first is the training error, that is, the error on the training set. The second quantity is a function of the Vapnik–Chervonenkis (VC) dimension [53.1] and is called the VC bound. The underlying theory shows that a learning machine with a low VC bound will have a low generalization error. Thus, a low-VC classifier will have a low CE on new data. Similarly, a prediction model with a low VC bound will have a low MMRE or a high PRED(25)

when used for future predictions. Consider the classification problem for a set of separable patterns. For this case, SVM derives a learning machine with a training error of zero. It also minimizes the second term.

However, in practice, the VC dimension of nonlinear classifiers or predictors, such as radial basis functions, cannot be accurately estimated and hence the results of this theory cannot be precisely implemented. That is why the statement above about SVM used the words approximate implementation. What is discussed in subsequent sections is a practical realization of this theory. We first show how the SVM develops classifiers for the separable case. SVM algorithms are then derived to deal with the more difficult cases of nonseparable patterns. These ideas are then extended to nonlinear prediction models.

53.4 Linearly Separable Patterns

This is the simplest classification task and was the first to be introduced for support vector machines (SVM) [53.1]. It is applicable for data that are linearly separable in the input space. Although not very useful for many real-world applications, it helps focus on the key aspects for developing SVM classifiers for more complex applications. We are given training data as in (53.1). We seek to determine a decision function D such that $D(\boldsymbol{x}) = y$, where y is the class label of data point $\boldsymbol{x}$. The goal is to find a D that minimizes generalization error. The decision boundary is given by the hyperplane

$$D(\boldsymbol{x}) : \boldsymbol{w}^T \boldsymbol{x} + b = 0 \,, \tag{53.5}$$

where $\boldsymbol{w}$ is a d-dimensional weight vector, and b is the bias term. If the data is separable, it will be correctly classified, i. e., for $i = 1, 2, \ldots, n$,

$$\begin{aligned} &\boldsymbol{w}^T \boldsymbol{x}_i + b \geq 0 \text{ for } y_i = +1 \\ \text{and} \quad &\boldsymbol{w}^T \boldsymbol{x}_i + b < 0 \text{ for } y_i = -1 \,. \end{aligned} \tag{53.6}$$

Separation between the decision function and the closest data point is called the *margin of separation* and equals $1/||\boldsymbol{w}||$. There can be many possible hyperplanes that separate the two classes. However, intuitively, the larger the margin, the lower the generalization error. This is so because points closest to the decision surface are harder to classify, and by providing a maximal separation between the two classes it should be easier to classify a new data point. For illustration, sample data points from two classes are shown in Fig. 53.2 with two margins, small and large. The boundaries with a larger margin can be seen to be preferable.

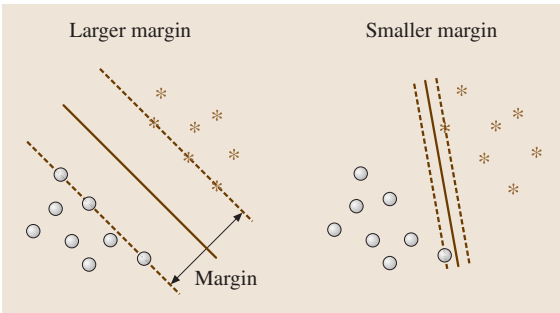


Fig. 53.2 Two linearly separable classes

53.4.1 Optimal Hyperplane

Given the linearly separable training data, the hyperplane that maximizes the margin is sought in the support vector machine formulation. Equivalently, we seek $(\boldsymbol{w}, b)$, which solves the following problem:

$$L\,(\mathrm{w}) = \min_{w\,b} \left(\frac{1}{2} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w} \right) \,. \tag{53.7}$$

$$\text{subject to} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b \right) \geq 1, \quad i = 1,\ 2, \ldots,\ n \,. \tag{53.8}$$

The aforementioned optimization problem may be solved by the method of Lagrange multipliers. Usually, the dual of this primal problem is solved. For details, see [53.3, 4]. The dual problem can then be stated as finding the Lagrange multipliers for the data in (53.1) that maximize the following objective function:

$$L_{\mathrm{D}} \equiv \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,\,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \cdot \boldsymbol{x}_j \,, \tag{53.9}$$

The constraints are

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \,, \tag{53.10}$$

$$\alpha_i \geq 0, \quad i = 1,\ 2, \ldots,\ n \,. \tag{53.11}$$

Note that in the dual only the input data appears in (53.9) in the form of dot products of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $i, j = 1, 2, \ldots, n$.

The solution yields the Lagrange multipliers α_i^*, $i = 1, 2, \ldots, n$. The optimal weight vector and the bias term are then given by

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i \tag{53.12}$$

$$\text{and } \boldsymbol{w}^{*\mathrm{T}} \boldsymbol{x} + b \geq 1 \,. \tag{53.13}$$

The optimal decision hyperplane can now be obtained by substituting for $\boldsymbol{w}^*$ and b^* in (53.5) as

$$D(x) : \boldsymbol{w}^{*\mathrm{T}} \cdot \boldsymbol{x} + b^* = 0 \,. \tag{53.14}$$

To classify a new data point, only its $\boldsymbol{x}$ will be known, and its y is determined based on (53.12) and (53.14) as

$$\text{class of new data} = \operatorname{sgn}\left[\sum_{i=1}^{n} \alpha_i^* y_i \, (\boldsymbol{x}_i \cdot \boldsymbol{x}) + b^*\right] . \tag{53.15}$$

Note from (53.15) that only nonzero α_i^* participate in determining the class. The indices of these determine the data points that determine the class. These points are support vectors.

53.4.2 Relationship to the SRM Principle

To see how minimization in (53.7), or equivalently maximizing the margin, is related to implementing the SRM principle, suppose that the following bound holds,

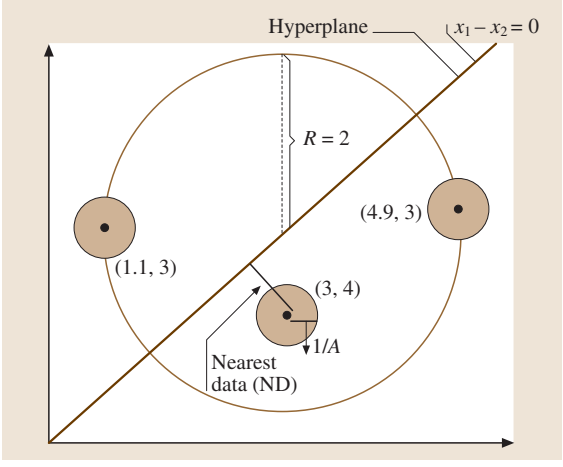


Fig. 53.3 Relationship between maximal margin and VC dimension

$||\boldsymbol{w}|| < A$, which means that the distance $d(\boldsymbol{w}, b; \boldsymbol{x})$ of a point from the hyperplane $(\boldsymbol{w}, b)$ is greater than or equal to $1/A$. The VC dimension, h, of the set of canonical hyper planes in n-dimensional space is bounded by

$$h \leq \min\left[R^2 A^2, n\right] + 1 \,,$$

where R is the radius of a hypersphere enclosing all the data points. Minimizing the above is equivalent to minimizing an upper bound on the VC dimension. Let us illustrate this by a simple numerical example.

Consider three ($n = 3$) data: (1.1, 3), (3, 4), and (4.9, 3), as shown in Fig. 53.3. The radius of a hypersphere enclosing all the data points, R, is 2. The distance between the nearest point (3, 4) and hyperplane $(x_1 - x_2) = 0$ is

$$\begin{aligned} d(\boldsymbol{w}, b; \boldsymbol{x}) &= \frac{\left|\boldsymbol{w} x_p \pm b\right|}{\|\boldsymbol{w}\|} \\ &= \frac{\left|w_1 x_{1p} + w_2 x_{2p} + \hbar\, w_n x_{np} \pm b\right|}{\sqrt{w_1^2 + w_2^2 + \hbar + w_n^2}} \\ &= \frac{|(1)\,(3) + (-1)\,(4)|}{\sqrt{1^2 + (-1)^2}} = 0.7071 \,. \end{aligned}$$

Therefore, $1/A \leq 0.7071 \approx 1.4142 \leq A$. In this example, we define $A = 1.5$ and $\min[R^2 A^2, n] + 1 = \min\left[\left(2^2\right)\left(1.5^2\right), 2\right] + 1 = \min[9, 2] + 1 = 3$. Therefore, the VC dimension, $h \leq 3$. Also, we can show that the norm of the weight vector should be equal to the inverse of the distance of the nearest point in the data set to the hyperplane. Here, the inverse of the distance between the nearest point (3, 4) and the hyperplane is $1/0.7071 = 1.4142$, which is equal to the norm of the weight vector $[1 - 1]$, $\sqrt{1^2 + (-1)^2} = 2 = 1.4142$.

53.4.3 Illustrative Example

We take a simple data set to illustrate in detail the computational steps involved in deriving the optimal separating hyperplane. Consider five normalized input data points in a two-dimensional plane along with their class labels, as shown in Table 53.1. Recall that our goal is to solve (53.9) subject to the constraints (53.10) and (53.11). We first compute the dot-product kernel of the input points and then compute $H(i, j) = y_i \cdot y_j \cdot \boldsymbol{x}_i^{\mathrm{T}} \cdot \boldsymbol{x}_j$

as shown below.

$$x_i^{\mathrm{T}} \cdot x_j = \begin{pmatrix} x_1^{\mathrm{T}}x_1 & x_1^{\mathrm{T}}x_2 & x_1^{\mathrm{T}}x_3 & x_1^{\mathrm{T}}x_4 & x_1^{\mathrm{T}}x_5 \\ \hbar & & \hbar & & \hbar \\ x_5^{\mathrm{T}}x_1 & x_5^{\mathrm{T}}x_2 & x_5^{\mathrm{T}}x_3 & x_5^{\mathrm{T}}x_4 & x_5^{\mathrm{T}}x_5 \end{pmatrix}$$

$$= \begin{pmatrix} \left(0\ 0\ \begin{pmatrix}0\\0\end{pmatrix}\right) & \left(0\ 0\ \begin{pmatrix}0\\1\end{pmatrix}\right) & \left(0\ 0\ \begin{pmatrix}0.3333\\0.75\end{pmatrix}\right) & \left(0\ 0\ \begin{pmatrix}0.6667\\0.75\end{pmatrix}\right) & \left(0\ 0\ \begin{pmatrix}1\\1\end{pmatrix}\right) \\ \hbar & & \hbar & & \hbar \\ \left(1\ 1\ \begin{pmatrix}0\\0\end{pmatrix}\right) & \left(1\ 1\ \begin{pmatrix}0\\1\end{pmatrix}\right) & \left(1\ 1\ \begin{pmatrix}0.3333\\0.75\end{pmatrix}\right) & \left(1\ 1\ \begin{pmatrix}0.6667\\0.75\end{pmatrix}\right) & \left(1\ 1\ \begin{pmatrix}1\\1\end{pmatrix}\right) \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \hbar & & \hbar & & \hbar \\ 0 & 1.0000 & 1.0833 & 1.4167 & 2.0000 \end{pmatrix},$$

$$H = \begin{pmatrix} y_1 & y_1 & \hbar & y_1 \\ \hbar & \hbar & \hbar & \\ y_5 & y_5 & \hbar & y_5 \end{pmatrix} .* \begin{pmatrix} y_1 & y_2 & & y_5 \\ \hbar & & \hbar & \hbar \\ y_1 & y_2 & & y_5 \end{pmatrix} .* \boldsymbol{x}_i^{\mathrm{T}} \cdot \boldsymbol{x}_j$$

$$= \begin{pmatrix} -1 & -1 & \hbar & -1 \\ \hbar & \hbar & \hbar & \\ -1 & -1 & \hbar & -1 \end{pmatrix} .* \begin{pmatrix} -1 & +1 & \hbar & -1 \\ \hbar & & \hbar & \hbar \\ -1 & +1 & \hbar & -1 \end{pmatrix} .* \begin{pmatrix} 0 & 0 & \hbar & 0 \\ \hbar & & \hbar & \hbar \\ 0 & 1.0000 & \hbar & 2.0000 \end{pmatrix},$$

$$\text{or } H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0.7500 & -0.7500 & -1.0000 \\ 0 & 0.7500 & 0.6736 & -0.7847 & -1.0833 \\ 0 & -0.7500 & -0.7847 & 1.0069 & 1.4167 \\ 0 & -1.0000 & -1.0833 & 1.4167 & 2.0000 \end{pmatrix}.$$

With the expression H as above, we now use the quadratic programming (QP) program, MATLAB Support Vector Machine Toolbox [53.5] to solve the dual problem given in (53.9), subject to the constraints in (53.10) and (53.11). The solution yields the optimal Lagrange multipliers given as

$$\left(\alpha^*\right)^{\mathrm{T}} = (7.11,\ 0.00,\ 32.22,\ 25.11,\ 0.00)\,.$$

Next, the optimal weight vector is obtained from (53.12) and its norm $||\boldsymbol{w}^*|| = \sqrt{64.44}$. Thus the maximal margin separating the hyperplanes is $2/||\boldsymbol{w}^*|| = 0.25$. Finally, the optimal bias from (53.13) is obtained as $b^* = -1.0$.

As mentioned before, only the nonzero values of the Lagrange multipliers participate in the solution, and they are $\alpha_1^* = 7.11, \alpha_3^* = 32.22, \alpha_4^* = 25.11$.

The input data points (1, 3, and 4) corresponding to the indices of these become the support vectors; the other two data points (2 and 5) can be ignored for classification decision. Thus the decision function of (53.15) can be written as

$$D(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{5} y_i \alpha_i^* \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x} + b^*\right). \qquad (53.16)$$

Next, we illustrate how a new data point for which only the $\boldsymbol{x}$ values are known is labeled by the above classifier. We first normalize the new data point according to the same scheme as used for the training data. Let the new point be (0.67, 1.00). The class computations proceed as follows. First, compute the dot product and H for the

Table 53.1 Data points for the illustrative example

X		Y
0.00	1.00	−1
0.00	1.00	+1
0.33	0.75	+1
0.67	0.75	−1
1.00	1.00	−1

new point and then substitute in (53.16), which yields the class as +1.

A graphical representation of the five data points used to develop the classifier, the optimal separating hyperplane, and the decision boundaries for classes −1 and +1 are shown in Fig. 53.4 in the normalized input space. We note that data points 1 and 4 are on the boundary for class −1, and data 3 is on the boundary of class +1. These are the three support vectors that participate in classification decisions. The other two points, 2 and 5, play no role in classification decisions.

53.5 Linear Classifier for Nonseparable Classes

In real-world applications, it is not realistic to construct a linear decision function without encountering errors. If the data are noisy, in general there will be no linear separation in the input space. Two situations may arise. In the first, data points fall in the region of separation, but on the right side so that the classification is correct. In the second case, data points fall on the wrong side and misclassification of points occurs. To accommodate such situations, the problem of the separable case is modified as follows. First, the classification constraints for the separable case are revised by adding slack variables (ξ_i). Next, the cost of constraint violation is set to C. With these modifications the function to be minimized becomes

$$L(\boldsymbol{w}, \xi_i) = \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + C\sum_{i=1}^{n}\xi_i \,, \tag{53.17}$$

$$\text{subject to } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1 - \xi_I \tag{53.18}$$

$$\text{and } \xi_i \geq 0, i = 1, 2, \ldots, n \,. \tag{53.19}$$

Minimizing $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ is related to minimizing the Vapnik–Chervonenkis (VC) dimension of the SVM, and the second term in (53.17) is an upper bound on the number of test errors. Thus, C controls the tradeoff between maximum margin and classification error, and ξ_i measures the deviation of a data point from the ideal condition of separability. If $0 \leq \xi_i \leq 1$, the data point is inside the decision region, but on the right side. If $\xi_i > 1$, the data point is on the wrong side. The data points that satisfy the above constraints are the support vectors. Proceeding as for the separable case with Lagrange multipliers α_i, the new dual problem is to

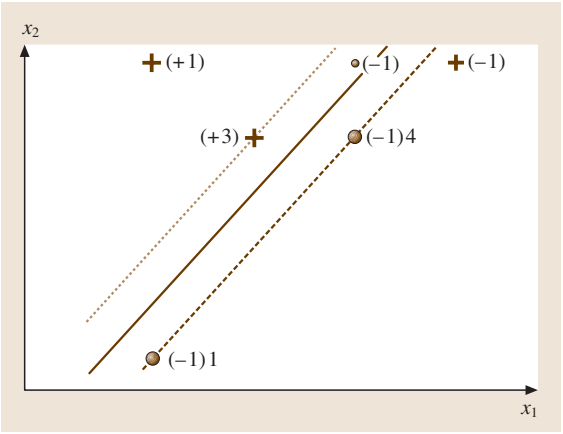


Fig. 53.4 Graphical representation of illustrative example

maximize

$$L(\alpha) = \sum \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \tag{53.20}$$

$$\text{subject to } \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{53.21}$$

$$\text{and } 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, n \,. \tag{53.22}$$

Then the optimal weights are

$$\boldsymbol{w}^* = \sum_{i=1}^{n}\alpha_i^* y_i \boldsymbol{x}_i \,. \tag{53.23}$$

Again, only those points with nonzero α_i^* contribute to $\boldsymbol{w}^*$, i. e., only those points which are the support vectors.

53.6 Nonlinear Classifiers

This situation arises when the data are linearly nonseparable in the input space and the separation lines are nonlinear hypersurfaces. This is a common case in practical applications. Here, we seek nonlinear decision boundaries in the input space. For this, the approach described above is extended to derive nonlinear decision

boundaries. This is achieved by first mapping the input data into a higher-dimensional feature space using an inner-product kernel $K(\boldsymbol{x}, \boldsymbol{x}_i)$. Then, an optimum separating hyperplane is constructed in the feature space. This hyperplane is defined as a linear combination of the feature space vectors and solves a linear classification problem in the feature space. Together, these two steps produce the solution for the case of nonlinear classifiers.

53.6.1 Optimal Hyperplane

We provide below a brief justification for the above two-step procedure. According to Cover's theorem [53.3, 6], a nonseparable space may be nonlinearly transformed into a new feature space where the patterns are very likely separable. Three inner-product kernels employed for SVMs are listed in Table 53.2. Among these, the Gaussian kernel is most commonly used in practical applications.

Finally, the dual of the constrained optimization problem for this case can be obtained as

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), \tag{53.24}$$

where $Q(\alpha)$ has to be maximized with respect to the α_i subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{53.25}$$

$$\text{and } 0 \le \alpha_i \le C, i = 1, 2, \ldots, n . \tag{53.26}$$

Here the parameter C is to be specified by the user. In the above, $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the ij-th element of the symmetric $n \times n$ matrix $\mathbf{K}$. A solution of the above problem yields the optimum Lagrange multipliers α_i^*, which yield the optimal weights as

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{\phi}(\boldsymbol{x}_i) , \tag{53.27}$$

where $\boldsymbol{\phi}(\boldsymbol{x}_i)$ represents the mapping of the input vector $\boldsymbol{x}_i$ into the feature space.

53.6.2 Illustrative Example

We illustrate the development of nonlinear classifiers using a Gaussian kernel for a small data set. The step-by-step solution procedure can be stated as follows:

- Preprocess input data
- Specify C, the kernel and its parameter
- Compute the inner-product matrix H
- Perform optimization using quadratic programming and compute the optimum α^*
- Compute the optimum weight vector $\boldsymbol{w}^*$
- Obtain support vectors and the decision boundary

Consider a data set of five points that consists of a 5×2 input matrix and a 5×1 vector y. Here, $\boldsymbol{x}_1$ is the normalized fan-in, $\boldsymbol{x}_2$ is the normalized module size in lines of code, and y is the module class.

$$\begin{array}{cc} X & y \\ \left(\begin{array}{cc:c} 0.29 & 0.00 & +1 \\ 1.00 & 0.02 & -1 \\ 0.00 & 0.19 & -1 \\ 0.06 & 1.00 & +1 \\ 0.02 & 0.17 & -1 \end{array}\right) & \end{array}$$

We use the radial basis function (rbf) kernel with $\sigma = 1.3$, and C is taken to be 100. Note that selection of C and σ is an important problem, as will be discussed later. These values are selected for illustrative purpose and are based on some preliminary analysis.

Next, the matrix H is obtained as

$$H(i, j) = \sum_{i=1}^{n} \sum_{j=1}^{n} y_i * y_j * \text{kernel}\left(rbf, \boldsymbol{x}_i, \boldsymbol{x}_j\right)$$
$$\equiv (5 \times 5)^* . (5 \times 5)^* . (5 \times 5) .$$

Table 53.2 Three common inner-product kernels

Type	$K(\boldsymbol{x}, \boldsymbol{x}_i), i = 1, 2, \ldots, n$	Comments
Linear	$\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}_i$	
Polynomial	$(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}_i + 1)^b$	b is user-specified
Gaussian	$\exp\left[-(1/2\sigma^2)(\|\|\boldsymbol{x} - \boldsymbol{x}_i\|\|^2)\right]$	σ^2 is user-specified

By substituting the appropriate values, the computations for H proceed as shown below. Here the symbol $\hbar$ represents entries not shown but obtained similar to the shown entries.

$$
\begin{aligned}
H &= \begin{pmatrix} y_1 & \hbar & y_1 \\ y_2 & \hbar & y_2 \\ y_3 & \hbar & y_3 \\ y_4 & \hbar & y_4 \\ y_5 & \hbar & y_5 \end{pmatrix} .* \begin{pmatrix} y_1 & y_2 & y_3 & y_4 & y_5 \\ \hbar & \hbar & \hbar & \hbar & \hbar \\ y_1 & y_2 & y_3 & y_4 & y_5 \end{pmatrix} \\
&\quad .* \begin{pmatrix} rbf(\sigma), x_1, x_1 & rbf(\sigma), x_1, x_2 & rbf(\sigma), x_1, x_3 & rbf(\sigma), x_1, x_4 & rbf(\sigma), x_1, x_5 \\ rbf(\sigma), x_2, x_1 & rbf(\sigma), x_1, x_2 & rbf(\sigma), x_1, x_3 & rbf(\sigma), x_1, x_4 & rbf(\sigma), x_1, x_5 \\ rbf(\sigma), x_3, x_1 & rbf(\sigma), x_1, x_2 & rbf(\sigma), x_1, x_3 & rbf(\sigma), x_1, x_4 & rbf(\sigma), x_1, x_5 \\ rbf(\sigma), x_4, x_1 & rbf(\sigma), x_1, x_2 & rbf(\sigma), x_1, x_3 & rbf(\sigma), x_1, x_4 & rbf(\sigma), x_1, x_5 \\ rbf(\sigma), x_4, x_1 & rbf(\sigma), x_1, x_2 & rbf(\sigma), x_1, x_3 & rbf(\sigma), x_1, x_4 & rbf(\sigma), x_1, x_5 \end{pmatrix} .* \\
&= \begin{pmatrix} 1 & \hbar & 1 \\ -1 & \hbar & -1 \\ -1 & \hbar & -1 \\ 1 & \hbar & 1 \\ -1 & \hbar & -1 \end{pmatrix} .* \begin{pmatrix} 1 & -1 & -1 & 1 & -1 \\ \hbar & \hbar & \hbar & \hbar & \hbar \\ 1 & -1 & -1 & 1 & -1 \end{pmatrix} \\
&\quad .* \begin{pmatrix} \exp\left(-\frac{\left\|\begin{pmatrix}0.29\\0.00\end{pmatrix}-\begin{pmatrix}0.29\\0.00\end{pmatrix}\right\|^2}{2(1.3)^2}\right) & \hbar & \exp\left(-\frac{\left\|\begin{pmatrix}0.29\\0.00\end{pmatrix}-\begin{pmatrix}0.02\\0.17\end{pmatrix}\right\|^2}{2(1.3)^2}\right) \\ \hbar & \hbar & \hbar \\ \exp\left(-\frac{\left\|\begin{pmatrix}0.02\\0.17\end{pmatrix}-\begin{pmatrix}0.29\\0.00\end{pmatrix}\right\|^2}{2(1.3)^2}\right) & \hbar & \exp\left(-\frac{\left\|\begin{pmatrix}0.02\\0.17\end{pmatrix}-\begin{pmatrix}0.02\\0.17\end{pmatrix}\right\|^2}{2(1.3)^2}\right) \end{pmatrix} \\
&= \begin{pmatrix} 1 & -0.86 & -0.97 & 0.73 & -0.97 \\ -0.86 & 1 & 0.74 & -0.58 & 0.75 \\ -0.97 & 0.74 & 1 & -0.82 & 1.00 \\ 0.73 & -0.58 & -0.82 & 1 & -0.82 \\ -0.97 & 0.75 & 1.00 & -0.82 & 1 \end{pmatrix} .
\end{aligned}
$$

After performing optimization by quadratic programming, the optimal α values are obtained. Then the optimal vector $\boldsymbol{w}$ is computed from α^* and H. Its squared length is computed from these as below, where $\alpha^{*\prime}$ is the transpose of α^*

$$
\begin{aligned}
w^2 &= \alpha^{*\prime} * H * \alpha^* \\
&= \begin{pmatrix} 100 & 29.40 & 0 & 20.60 & 92.90 \end{pmatrix} \\
&\quad * \begin{pmatrix} 1 & -0.86 & -0.97 & 0.73 & -0.97 \\ -0.86 & 1 & 0.74 & -0.58 & 0.75 \\ -0.97 & 0.74 & 1 & -0.82 & 1.00 \\ 0.73 & -0.58 & -0.82 & 1 & -0.82 \\ -0.97 & 0.75 & 1.00 & -0.82 & 1 \end{pmatrix} * \begin{pmatrix} 100 \\ 29.40 \\ 0 \\ 20.60 \\ 92.90 \end{pmatrix} \\
&= 102.
\end{aligned}
$$

The indices of the support vectors are those α^* that satisfy $0 < \alpha^* \leq C$. Here, $\alpha_3^* = 0$ so that points 1, 2, 4, and 5 become the support vectors for this classification problem; point 3 plays no role and can be ignored.

A graphical illustration of the decision boundaries in the input space is shown in Fig. 53.5. Also, note that the decision boundaries are nonlinear, while the decision hyperplane in the feature space computed from the feature vector is expected to be linear. Finally, for this problem, data point 1 is misclassified as being in class -1 rather than in class $+1$, i. e., the classification error of the model derived here is 20%. To classify a new point, suppose its normalized values using the same normalization as for the training data are

(0.03, –0.03). The classification computations proceed as follows:

$$
\begin{aligned}
H(i,j) &= \sum_{i=1}^{m}\sum_{j=1}^{n} y_j * \text{kernel}\left(rbf, xt_i, x_j\right) \\
&= \begin{pmatrix} y_1 & y_2 & y_3 & y_4 & y_5 \end{pmatrix} \\
&\quad * \begin{pmatrix} (rbf(\sigma), xt_1, x_1) \\ (rbf(\sigma), xt_1, x_2) \\ (rbf(\sigma), xt_1, x_3) \\ (rbf(\sigma), xt_1, x_4) \\ (rbf(\sigma), xt_1, x_5) \end{pmatrix}^{\top} \\
&= \begin{pmatrix} 1 & -1 & -1 & 1 & -1 \end{pmatrix} \\
&\quad .* \begin{pmatrix} \exp\left(-\frac{\left\| \begin{pmatrix} 0.03 \\ -0.03 \end{pmatrix} - \begin{pmatrix} 0.29 \\ 0.00 \end{pmatrix} \right\|^2}{2.(1.3)^2}\right) \\ \hbar \\ \exp\left(-\frac{\left\| \begin{pmatrix} 0.03 \\ -0.03 \end{pmatrix} - \begin{pmatrix} 0.02 \\ 0.17 \end{pmatrix} \right\|^2}{2.(1.3)^2}\right) \end{pmatrix}^{\top} \\
&= \begin{pmatrix} 0.98 & -0.76 & -0.98 & 0.73 & -0.99 \end{pmatrix} ; \\
y &= \text{sgn}\left(H * \alpha + \text{bias}\right)
\end{aligned}
$$

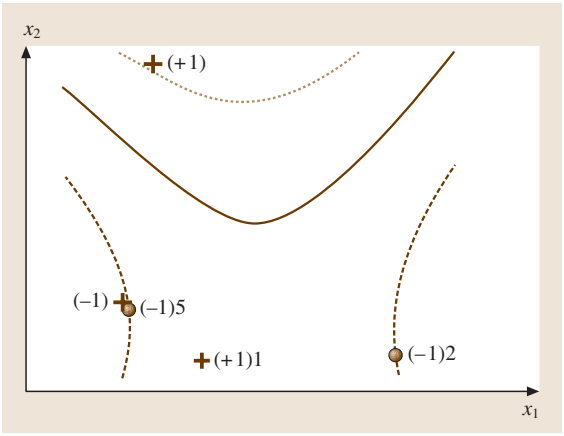


Fig. 53.5 Graphical illustration of nonlinear decision boundaries

$$
\begin{aligned}
&= \text{sgn}\left(\begin{pmatrix} 0.98 \\ -0.76 \\ -0.98 \\ 0.73 \\ -0.99 \end{pmatrix}^{\top} * \begin{pmatrix} 100.00 \\ 29.40 \\ 0.00 \\ 20.60 \\ 92.90 \end{pmatrix} + 0 \right) \\
&= \text{sgn}\left(-0.92\right) = -1 .
\end{aligned}
$$

The new module belongs to class –1.

53.7 SVM Nonlinear Regression

As mentioned earlier, the support vector technique was initially developed for classification problems. This approach has been extended to nonlinear regression where the output y are real-valued. A general nonlinear regression model for y based on $\boldsymbol{x}$ can be written as

$$y = f(\boldsymbol{x}, \boldsymbol{w}) + \delta \,, \tag{53.28}$$

where f represents a function, $\boldsymbol{w}$ is a set of parameters, and δ represents noise. In terms of some nonlinear basis functions, as discussed earlier, we can write $\hat{y}$, an estimate of y, as

$$\hat{y} = \boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x}) \,. \tag{53.29}$$

Next, we employ the commonly used Vapnik's ε-loss function and estimate $\hat{y}$ via support vector regression as summarized below. The ε-loss is defined as

$$\text{Loss}_{\varepsilon}\left(y, \hat{y}\right) = \begin{cases} |y - \hat{y}| - \varepsilon, & \text{if } |y - \hat{y}| \geq \varepsilon \,. \\ 0 \quad\quad\quad , & \text{otherwise} \end{cases}$$

The dual problem for regression can be formulated using an approach similar to that for classification. The optimization problem now is to maximize Q:

$$
\begin{aligned}
Q\left(\alpha_i, \alpha_i'\right) &= \sum_{i=1}^{n} y_i \left(\alpha_i - \alpha_i'\right) - \varepsilon \sum_{i=1}^{n} \left(\alpha_i + \alpha_i'\right) \\
&\quad - \frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\alpha_i - \alpha_i'\right) \\
&\quad \times \left(\alpha_j - \alpha_j'\right) K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)
\end{aligned} \tag{53.30}
$$

$$\text{subject to} \quad \sum_{i=1}^{n} \left(\alpha_i - \alpha_i'\right) = 0 \,, \tag{53.31}$$

$$0 \le \alpha_i, \alpha_i' \le C, i = 1, 2, \ldots, n \,. \tag{53.32}$$

In (53.30–53.32), ε and C are user-specified values. The optimal values of α_i and α_i' are used to find the optimal value of the weight vector. The estimated $\hat{y}$ is given by

$$\hat{y}\,(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = \sum_{i=1}^{n} \left(\alpha_i - \alpha_i'\right) K\,(\boldsymbol{x}, \boldsymbol{x}_i) \,.$$

53.8 SVM Hyperparameters

The classification and regression modeling problems using SVM are formulated as quadratic programming (QP) optimization problems. Many algorithms are available for solving QP problems and are commonly used for support vector modeling applications. However, there are some parameters that are to be specified by the user. These are called *the SVM hyperparameters* and are briefly described below.

For the linearly nonseparable case, we need to specify the penalty parameter C. It controls the tradeoff between the small function norm and empirical risk minimization. Further, for nonlinear classifiers, we also need to specify the kernel and its parameters. For example, for the radial basis function kernel, its width, σ, needs to be specified by the user. In practical applications, there are no easy answers for choosing C and σ. In general, to find the best values, different combinations are tried and their performances are compared, usually via an independent data set, known as the validation set. However, some empirical rules for their determination have been proposed in the literature [53.7, 8].

For nonlinear regression, a loss function is specified. In support vector machine applications, a commonly used loss function is the so-called *ε-loss function* as indicated above. Thus, for regression problems this additional hyperparameter is to be specified by the user. Generally, a trial-and-error approach is used to evaluate the performance of different hyperparameter combinations on some validation data set.

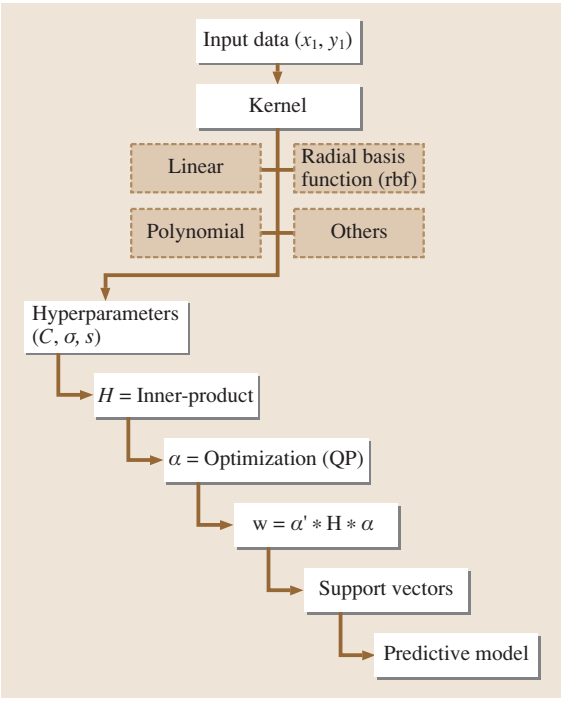


Fig. 53.6 Support vector modeling flow chart

53.9 SVM Flow Chart

A generic flow chart depicting the development of support vector classification and regression models is shown in Fig. 53.6. For given data ($\boldsymbol{x}$, y), the kernel is selected by the user, followed by the appropriate hyperparameters. Computation of several intermediate quantities and optimization by quadratic programming yields the weights and support vectors. Finally, these values are used to define the classification or regression model.

53.10 Module Classification

There are several reasons for developing software classification models. One is the limited availability of resources. Not all modules can be treated in the same way. The potentially critical modules require a more time-consuming and costly development process that involves activities such as more rigorous design and code reviews, automated test-case generation and unit testing, etc. Another reason is that faults found later in the development life-cycle are more expensive to correct than those found earlier.

The problem of software module classification has been addressed in the software engineering literature for more than thirty years. Different techniques have been applied by many authors with varying degrees of predictive accuracy [53.9, 10]. Most of the early work on this topic used statistical techniques such as discriminant analysis, principle-component analysis, and factor analysis, as well as decision or classification trees [53.11]. In recent years, machine learning techniques and fuzzy logic have also been used for software module classification. Typical of these are classification and regression trees (CART), case-based reasoning (CBR), expert judgment, and neural networks [53.10]. The main problem with most of the current models is their low predictive accuracy. Since the published results vary over a wide range, it is not easy to give a specific average accuracy value achieved by current models.

In this section, we develop support vector classification models for software data obtained from the public-domain National Aeronautics and Space Administration (NASA) software metrics database [53.12]. It contains several product and process metrics for many software systems and their subsystems. The fifteen module-level metrics used here and a brief description of each are given in Table 53.3. The metric x_7 is the number of faults, while x_9 to x_{22} are module-level product metrics which include the design-based metrics (x_9, x_{10}, x_{18}) and primarily coding metrics (x_{13}, x_{14}, x_{15}). Metrics x_{19} to x_{22} are Halstead's software science measures; x_{19} and x_{20} are the vocabulary, while x_{21} and x_{22} are measures of program size. Other metrics are self-explanatory. These represent typical metrics used in module classification studies. This system consists of 67 modules with a size of about 40k lines of code.

Table 53.3 List of metrics from NASA database

X_7	Faults
X_9	Fan out
X_{10}	Fan in
X_{11}	Input–output statements
X_{12}	Total statements
X_{13}	Size of component in program lines
X_{14}	Number of comment lines
X_{15}	Number of decisions
X_{16}	Number of assignment statements
X_{17}	Number of formal statements
X_{18}	Number of input–output parameters
X_{19}	Number of unique operators
X_{20}	Number of unique operands
X_{21}	Total number of operators
X_{22}	Total number of operands

Here, faults are the outputs and the others are the inputs. Referring to (53.1), we have $n = 67$ and $d = 14$. We first preprocess the input data. After transformation, this data set resides in a fourteen-dimensional unit cube. To determine module class, we use a threshold value of 5 so that, if $x_7 \leq 5$, the class is -1, and $+1$ otherwise.

We now develop nonlinear classifiers for this data set using the SVM algorithm of Sect. 53.6 [53.13]. The

Table 53.4 Classification results

Set	σ	C	SV	Classification error (average)	
				Training	5CV
I	0.8	1	30.6	0.18	0.21
II	1.2	100	26.8	0.10	0.20
III	1.6	1000	27.2	0.12	0.20
IV	2.0	100	26.6	0.13	0.21
V	2.4	1000	25.8	0.09	0.21
VI	2.8	1000	26.4	0.11	0.19
VII	3.2	1000	25.8	0.12	0.17
VIII	3.6	1000	26.2	0.12	0.17
IX	4.0	1000	25.4	0.12	0.21

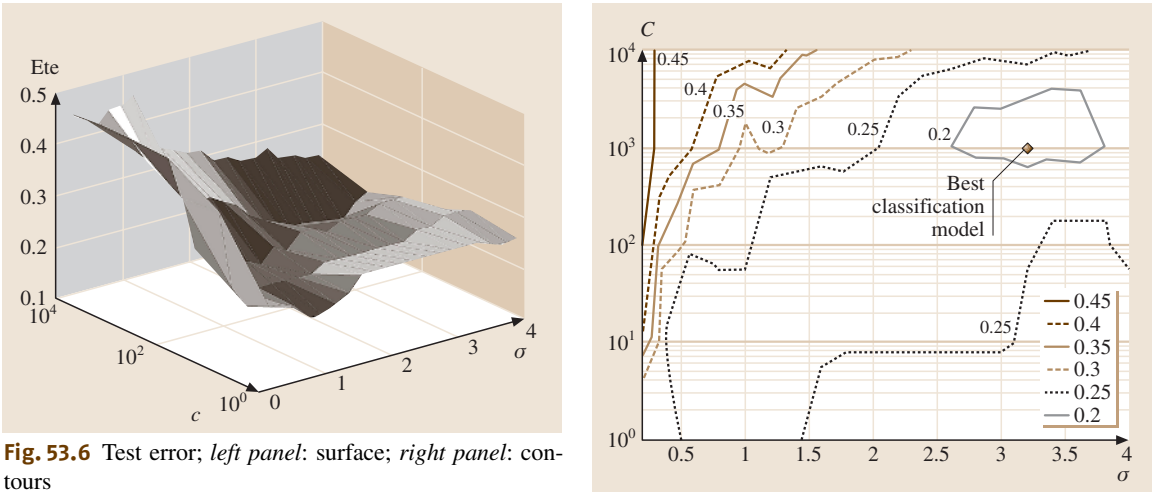


Fig. 53.6 Test error; *left panel*: surface; *right panel*: contours

optimization problem to be solved is given by (53.24–53.26). First, we choose a kernel. Since Gaussian is a popular choice, we also choose it. Next we need to specify the hyperparameters, that is, the width of the Gaussian and the penalty parameter C. To determine the best combination of these two, we follow the common practice of performing a grid search. In this case the grid search is on the two parameters C and σ. We took $C = 10^{-2}(10)10^4$ and $\sigma = 0.8(0.4)4.0$ for a total of 56 grid points. For each combination, we use the SVM algorithm for nonlinear classifiers of Sect. 53.6. Further, we used five-fold cross validation (5CV) as a criterion for selecting the best hyperparameter combination. Thus, we are essentially doing a search for the best set of hyperparameters among the 56 potential candidates by constructing $56 \times 5 = 280$ classifiers, using the approach described in Sect. 53.6.

As an example, a list of nine sets and their corresponding classification errors is given in Table 53.4. The average number of input points that became support vectors for the given model is also included. Data such as that in Table 53.4 is used to select the best set.

From this we select set VII with $C = 1000$ and kernel width 3.2. To further study the behavior of the five-fold cross validation test error versus (C, σ), its surface and the contours are shown in Fig. 53.6. We note that the surface for this data is relatively flat for high (σ, C) values and sharp for low σ and high C. This behavior is quite typical for many applications. Also shown in the contour plot is the chosen set with $\sigma = 3.2$ and $C = 1000$. Finally, these values are used as the SVM hyperparameters to solve the optimization problem of (53.24–53.26) using quadratic programming. This gives the desired, possibly best, classification model for this data set. The developed model is likely to have an error of about 17% on future classification tasks.

53.11 Effort Prediction

Development of software effort-prediction models has been an active area of software engineering research for over 30 years. The basic premise underlying these models is that historical data about similar projects can be employed as a basis for predicting efforts for future projects. For both engineering and management reasons, an accurate prediction of effort is of significant importance in software engineering, and improving estimation accuracy is an important goal of most software development organizations. There is a continuing search for better models and tools to improve predictive performance.

The so-called general-purpose models are generally algorithmic models developed from a relatively large collection of projects that capture a functional relationship between effort and project characteristics [53.14, 15]. The statistical models are derived from historical data using statistical methods, mostly regression analysis. The statistical models were some of the earliest to be developed. Exam-

Table 53.5 Performance of effort prediction models

Data set	LooMMRE		LooPred(25)	
	expected training	expected test	average training	average test
D	0.54	0.52	0.33	0.46
$D-1$	0.28	0.35	0.72	0.54
$D-2$	0.24	0.24	0.72	0.64
$D-3$	0.12	0.25	0.44	0.80

ples of such models are the meta-model [53.16] and the MERMAID [53.17]. Recently, machine learning techniques have been used for software effort prediction modeling. These include neural networks [53.18], rule induction, and case-based reasoning.

The effort-modeling problem can be restated as follows from Sect. 53.2 above. We are given data about n projects $\{\boldsymbol{x}_i, y_i\} \in \mathbb{R}^n \times \mathbb{R}$, $i = 1, \ldots, n$, each consisting of d software features the y are the effort values. A general nonlinear regression model for y based on $\boldsymbol{x}$ was given in (53.28).

In this section we summarize the prediction model development by support vector nonlinear regression from [53.13]. The effort data and the project features are taken from [53.19]. The data were collected from a Canadian software house. It consists of 75 projects developed in three different environments. The data is grouped by each environment ($D-1$, $D-2$, $D-3$) and as combined projects (D). There were six features collected for each project. Thus, for data set D, $n = 75$ and $d = 6$ in (53.2).

The methodology for developing SVM nonlinear regression models is very similar to that used for module classification in Sect. 53.10. The optimization problem to be solved now is given in (53.30–53.32). Further, assuming a Gaussian kernel, three hyperparameters have to be specified. Therefore, a three-dimensional grid search has to be performed for selecting σ, C, and ε. The criterion for this selection can be MMRE or Pred(25). Note that we seek a low MMRE error and a high Pred(25) accuracy.

The final results of SVM modeling for the above data sets are summarized in Table 53.5 for both selection criteria. For projects in D, the best model obtained has test values of LooMMRE $= 0.52$ and LooPred(25) $= 0.46$. However, for $D-1$, $D-2$ and $D-3$, model performance is much better due to the fact that the projects in these data sets were developed in more homogeneous environments than those in D.

53.12 Concluding Remarks

We have presented an introduction to support vector machines, their conceptual underpinnings and the main computational techniques. We illustrated the algorithmic steps via examples and presented a generic SVM flowchart. Results from two software engineering case studies using SVM were summarized.

SVM is a very active area of research and applications. An impressive body of literature on this topic has evolved during the last decade. Many open problems, theoretical and applied, are currently being pursued. These include hyperparameter selection, Bayesian relevance vector machines, reduced SVM, multiclass SVM, etc. Applications include intrusion detection, data mining, text analysis, medical diagnosis and bioinformatics. Papers on these aspects regularly appear in the machine learning and related literature.

For further reading, chapters in *Kecman* [53.4], *Cherkassky* et al. [53.20] and *Haykin* [53.3] provide good insights. Books on SVM include *Cristianini* et al. [53.21], *Scholkopf* et al. [53.22], and *Vapnik* [53.1]. Tutorials, such as *Burges* [53.23], and other useful information is available at websites dealing with support vector machines and kernel machines. Software packages are also available from several websites.

References

53.1 V. N. Vapnik: *Statistical Learning Theory* (Wiley, New York 1998)

53.2 V. N. Vapnik: An overview of statistical learning theory, IEEE Trans. Neural Netw. **10(5)**, 988–1000 (1999)

53.3 S. Haykin: *Neural Networks – A Comprehensive Foundation*, 2nd edn. (Prentice Hall, Upper Saddle River, NJ 1999)

53.4 V. Kecman: *Learning and Soft Computing* (MIT Press, Cambridge, MA 2001)

53.5 S. R. Gunn: *MATLAB Support Vector Machine Toolbox* (1998), http://www.isis.ecs.soton.ac.uk/resources/svminfo/

53.6 T. Hastie, R. Tibshirani, J. H. Friedman: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, Berlin Heidelberg New York 2001)

53.7 S. S. Keerthi, C-J. Lin: Asymptotic behaviors of support vector machines with Gaussian kernels, Neural Comput. **15**(7), 1667–1689 (2003)

53.8 O. Chapelle, V. Vapnik: *Model Selection for Support Vector Machines. Advances in Neural Information Processing Systems* (AT&T Labs-Research, Lyone 1999)

53.9 A. L. Goel: *Software Metrics Statistical Analysis Techniques and Final Technical Report* (U. S. Army, 1995)

53.10 T. M. Khoshgoftaar, N. Seliya: Comparative assessment of software quality classification techniques: An empirical case study, Empir. Softw. Eng. **9**, 229–257 (2004)

53.11 C. Ebert, T. Liedtke, E. Baisch: Improving Reliability of Large Software Systems. In: *Annals of Software Engineering*, Vol. 8, ed. by A. L. Goel (Baltzer Science, Red Bank, NJ 1999) pp. 3–51

53.12 NASA IV & V Metrics Data Program. http://mdp.ivv.nasa.gov/

53.13 H. Lim: Support Vector Parameter Selection Using Experimental Design Based Generating Set Search (SVEG) with Application to Predictive Software Data Modeling. Ph.D. Thesis (Syracuse Univ., New York 2004)

53.14 B. W. Boehm: *Software Engineering Economics* (Prentice Hall, New York 1981)

53.15 L. H. Putnam: A general empirical solution to the macro sizing and estimating problem, IEEE Trans. Softw. Eng. **4**, 345–361 (1978)

53.16 J. W. Bailey, V. R. Basili: A Meta-Model for Software Development Resource Expenditures, Proceedings of the 5th International Conference on Software Engineering, San Diego, CA (IEEE Press, Piscataway, NJ 1981) 107–116

53.17 P. Kok, B. A. Kitchenham, J. Kirakowski: The MERMAID Approach to Software Cost Estimation. In: *Proceedings ESPRIT Technical Week* (Kluwer Academic, Brussels 1990)

53.18 M. Shin, A. L. Goel: Empirical data modeling in software engineering using radial basis functions, IEEE Trans. Softw. Eng. **36**(5), 567–576 (2000)

53.19 J. M. Desharnais: *Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Fonction. MSc Thesis* (Univ. of Quebec, Montreal 1988)

53.20 V. Cherkassky, F. Mulier: *Learning from Data: Concepts, Theory, and Methods* (Wiley-Interscience, New York 1998)

53.21 N. Cristianini, J. Shawe-Taylor: *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* (Cambridge Univ. Press, Cambridge 2000)

53.22 P. Scholkopf, A. Smola: *Learning with Kernels* (MIT Press, Cambridge, MA 2002)

53.23 C. J. C. Burges: A Tutorial on SVM for Pattern Recognition, Data Mining and Knowledge Discovery **2**, 167–212 (1998)

54. Optimal System Design

The first section of this chapter describes various applications of optimal system design and associated mathematical formulations. Special attention is given to the consideration the randomness associated with system characteristics. The problems are described from the reliability engineering point of view. It includes a detailed state-of-the-art presentation of various spares optimization models and their applications.

The second section describes the importance of optimal cost-effective designs. The detailed formulations of cost-effective designs for repairable and nonrepairable systems are discussed. Various cost factors such as failure cost, downtime cost, spares cost, and maintenance cost are considered. In order to apply these methods for real-life situations, various constraints including acceptable reliability and availability, weight and space limitations, and budget limitations are addressed.

The third section describes the solution techniques and algorithms used for optimal system-design problems. The algorithms are broadly classified as exact algorithms, heuristics, meta-heuristics, approximations, and hybrid methods. The merits and demerits of these algorithms are described. The importance of bounds on the optimal solutions are described.

The fourth section describes the usefulness of hybrid methods in solving large problems in a realistic time frame. A detailed description of the latest research findings relating to hybrid methods and their computational advantages are provided. One of the major advantages of these algorithms is finding the near-optimal solutions as quickly as possible and improving the solution quality iteratively. Further, each iteration improves the search efficiency by reducing the search space as a result of the improved bounds. The efficiency of the proposed method is demonstrated through numerical examples.

54.1 Optimal System Design

In everyday life, we come across various kinds of decision-making problems, ranging from personal decisions related to investment, travel, and career development to business decisions related to procuring equipment, hiring staff, product design, and modifications to existing design and manufacturing procedures. Decision analysis involves the use of a rational process for selecting the best of several alternatives. The solution to decision-making problem requires the identification of three main components.

1. What are the decision alternatives?
   Examples: Should I select vendor X or vendor Y? Should I keep an additional spare component or not?
2. Under what restrictions (constraints) is the decision to be made?

Examples: Do not spend more than $ 10 000 for procuring new equipment. On average, downtime of the system should not exceed two days in a year.

3. What is an appropriate objective criterion for evaluating the alternatives?
   Examples: Overall profit is maximum. Overall availability is maximum. Overall cost is minimum.

Generally, the alternatives of the decision problem may take the form of unknown variables. The variables are then used to construct the restrictions and the objective criterion in appropriate mathematical functions. The end results is a mathematical model relating the variables, constraints, and objective function. The solution of the model then yields the values of the decision variables that optimize (maximize or minimize) the value of the objective function while satisfying all the constraints [54.1]. The resulting solution is referred to as the optimum feasible solution (or simply optimal solution). A typical mathematical model for optimal decision-making is organized as follows:

Maximize or minimize (objective function)
subject to (constraints) .

In the mathematical models for optimization, the decision variables may be integer or continuous, and the objective and constraint functions may be linear and nonlinear. It should be noted that the discrete variables, such as component reliability choices that are restricted to the set {0.6, 0.75, 0.9}, as well as categorical variables such as names can be converted into equivalent integer variables (by using mapping). The optimization problems posed by these models give rise to variety of problem formulations. Each category of problem formulation (or the model) can be solved using a class of solution methods, each designed to account for the special mathematical properties of the model.

1. Linear programming problem: where all objective and constraint functions are linear, and all the variables are continuous.
2. Linear integer programming: is a linear programming problem with the additional restriction that all the variables are integers.
3. Nonlinear programming: where the objective or at least one constraint function is nonlinear, and all the variables are continuous.
4. Nonlinear integer programming: is a nonlinear programming problem with the additional restriction that all the variables are integers.
5. Nonlinear mixed integer programming: is a nonlinear programming problem where some variables are integers and other variables are continuous.

Problems with integer variables and nonlinear functions are difficult to solve. The difficulty further increases if there exist both integer and continuous variables.

54.1.1 System Design

System design is one of the important applications of optimal decision-making problems. One of the important goals of system design is to build the system such that it performs its functions successfully. When a system is unable to perform its functions, this is called a system failure. Several factors related to system design as well as external events influence the system functionality. In most cases, the effects of these factors are random, which means that they cannot be determined precisely but can only be explained through probability distributions. Therefore, failure events or the time to system failure are random variables. The engineering discipline that deals with the successful and unsuccessful (failure) operations of systems is known as reliability engineering. Reliability is one of the important system characteristics and is defined as the probability that the system performs its intended (or specified) functions successfully over a specified period of time under the specified environment. One of the goals of reliability engineering is to illustrate how high reliability, through careful design and analysis, can be built into systems within the limits of economical and physical constraints. Some important principles for enhancing system reliability are [54.2–4]:

1. Keep the system as simple as compatible with performance requirements. This can be achieved by minimizing the number of components in series and their interactions.
2. Increase the reliability of the components in the system. This can be achieved by reducing the variations in the components' strength and applied load (better quality control and controlled operational environment), increasing the strength of the components (better materials), and reducing the applied loads (derating). Alternatively, to some extent, this can be achieved by using large safety factors or management programs for product improvement.
3. Use burn-in procedures for components that have high infant mortality to eliminate early failures in the field.

4. Use redundancy (spares) for less-reliable components; this can be achieved by adding spares in the parallel or standby redundancy.
5. Use fault-tolerant design such that system can continue its functions even in the presence of some failures. This can be achieved using sparing redundancy, fault-masking, and failover capabilities.

In addition to this, if system or its components are repairable, the availability of the system should be considered as a system performance index. The availability of the system is the probability that the system is operational at a specified time. In the long run, the system availability estimate reaches an asymptotic value called the steady-state availability [54.5]. Therefore, in most cases, we focus our attention on improving the steady-state availability of the system. The system availability can be increasing by reducing the downtime. Some important principles for enhancing system reliability are:

1. Use methods that increase the reliability of the system.
2. Decrease the downtime by reducing delays in performing the repair. This can achieved by keeping additional spares on site, providing better training to the repair personnel, using better diagnosis procedures, and increasing the size of the repair crew.
3. Perform preventive maintenance such that components are replaced by new ones whenever they fail, or at some regular time intervals or age, whichever comes first.
4. Perform condition-based maintenance such that downtime related to either preventive or corrective maintenance is minimal.
5. Use better arrangement for exchangeable components.

Implementation of the above steps often consume some resources. The resources to improve system performance (reliability or availability) may be limited. This resources limitation may include available budget, space to keep components, and weight limitations. In such cases, the objective should be to obtain the maximum performance within the utilization of available resources. However, in some cases, achieving high performance may not lead to high profit (or low cost). In such cases, optimal designs should be performed to achieve the most cost-effective solution that strikes a balance between system failure cost and the cost of efforts for reducing the system failures.

54.1.2 System Design Objectives

Depending on the situation, the objective of optimal system design can be one of the following.

1. Maximize system performance.
   There are a number of measures that indicate the performance of a system. For nonrepairable systems, reliability is an important performance measure. For repairable systems, the availability or total uptime is important. When the system has several levels of performance (multi-state systems), the average capacity or throughput is important.
2. Minimize the losses associated with unwanted system behaviors.
   We can also design the system such that the losses associated with downtime can be minimized. Therefore, we can focus on reducing the unreliability, unavailability, downtime, or number of failures.
3. Maximize the overall profit (or minimize the overall cost) associated with the system.

In general, the optimal design corresponding to the maximum system performance (or minimum unwanted behavior) may not exactly coincide with the optimal design that maximizes system profit (or minimum overall cost). In such cases, the objective should be minimizing the overall cost associated with the system that meets the acceptable system performance as well as resource consumption.

In order to solve these optimization problems, we should specify the objectives in a mathematical form. Therefore, we should express the objective functions in a mathematical form.

54.1.3 Notation

m	number of subsystems in the system,
n_i	number of components in subsystem i; $i \in [1, \cdots, m]$,
k_i	minimum number of good components required for successful operation of the subsystem i; $1 \le k_i \le n_i$,
s_i	number of spares in subsystem i; $s_i = n_i - k_i$,
$\boldsymbol{n}$	vector of components; $\boldsymbol{n} = [n_1, \cdots, n_m]$,
$\boldsymbol{n^K}$	vector of components with $n_i = k_i$; $\boldsymbol{n^K} = [k_1, \cdots, k_m]$,
n^*	optimal value of n,
$n^{\mathrm{L}}, n^{\mathrm{U}}$	[lower, upper] bound on the optimal value of n; $n^{\mathrm{L}} \le n^* \le n^{\mathrm{U}}$; n_i^{L} and n_i^{U} are the

	lower and upper bounds on n_i^*; $\boldsymbol{n}^{\mathrm{L}}$ and $\boldsymbol{n}^{\mathrm{U}}$ are the lower and upper bounds on $\boldsymbol{n}^*$,
$\phi_i, \psi_i, \varphi_i$	[mean time to failure = MTTF, mean time to repair = MTTR, mean logistic delay time = MLDT] of a component in subsystem i,
γ_i	fixed miscellaneous cost per each repair of a component in subsystem i,
δ_i	variable cost per unit repair time for a component in subsystem i,
ν_i	frequency of failures for a component in subsystem i,
c_i	maintenance cost per unit time for each component in subsystem i,
c_{f}	cost of system downtime per unit time,
r_i or R_i	reliability of subsystem i,
Q_i	unreliability of subsystem i; $Q_i = 1 - R_i$,
R_{s}	reliability of the system,
Q_{s}	unreliability of the system; $Q_{\mathrm{s}} = 1 - R_{\mathrm{s}}$,
$f(.)$	is a function; $f(r_1, \cdots, r_m)$ is a function in r_i, which is used to represent system reliability in terms of component reliabilities,
$g_i(.)$	is a function; $g_i(r_1, \cdots, r_m)$ is a function in r_i, which is used to represent the constraints in terms of component reliabilities,
r_j^{l}	explicit lower limit on r_j,
r_j^{u}	explicit upper limit on r_j,
$T_{\mathrm{d}}(\boldsymbol{n})$	average downtime cost per unit time with component vector $\boldsymbol{n}$,
$T_{\mathrm{m}}(\boldsymbol{n})$	average maintenance cost per unit time with component vector $\boldsymbol{n}$,
$T(\boldsymbol{n})$	average system cost per unit time with component vector $\boldsymbol{n}$,
C_K	total maintenance cost with $\boldsymbol{n}^{\boldsymbol{K}}$; $C_K = T_{\mathrm{m}}(\boldsymbol{n}^{\boldsymbol{K}}) = \sum_{i=1}^{m} k_i c_i$
C_{U}	total maintenance cost at the upper bound $\boldsymbol{n}^{\mathrm{U}}$; $C_{\mathrm{U}} = T_{\mathrm{m}}(\boldsymbol{n}^{\mathrm{U}}) = \sum_{i=1}^{m} n_i^{\mathrm{U}} c_i$,
C_{L}	total maintenance cost at the lower bound $\boldsymbol{n}^{\mathrm{L}}$; $C_{\mathrm{L}} = T_{\mathrm{m}}(\boldsymbol{n}^{\mathrm{L}}) = \sum_{i=1}^{m} n_i^{\mathrm{L}} c_i$,
p_i, q_i	[availability, unavailability] or [reliability, unreliability] of a component in subsystem i,
A_i, U_i	[availability, unavailability] of subsystem i when there are n_i components in that subsystem; $A_i \equiv A_i(n_i)$; $U_i \equiv U_i(n_i)$,
$A(\boldsymbol{n}), U(\boldsymbol{n})$	[availability, unavailability] of the system with component vector $\boldsymbol{n}$,
h_i^1, h_i^0	Pr{system is operating | subsystem i is [operating, failed]},
binf$(k; p, n)$	cumulative distribution function of binomial distribution; binf$(k; p, n) \equiv \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$; binfc$(k; p, n) \equiv 1 - \mathrm{binf}(k; p, n)$,
gilb$(\cdot)$	greatest integer lower bound; floor$(\cdot)$

54.1.4 System Reliability

If the objective is maximization of system reliability, then we should express the system reliability in a mathematical form. The form of the reliability expression varies with system configuration.

Series Configuration

Series configuration is the simplest and perhaps one of the most common configurations. Let m be the number of subsystems (or components) in the system. In this configuration, all m subsystems (components) must be operating to ensure system operation. In other words, the system fails when any one of the m subsystems fails. Therefore, the reliability of the series system, when all subsystems are independent, is the product of reliabilities of its systems

$$R_{\mathrm{s}} = \prod_{i=1}^{m} R_i \,. \tag{54.1}$$

Parallel Configuration

In the parallel configuration, several paths (subsystems) perform the same operation simultaneously. Therefore, the system fails if all of the m subsystems fail. This is also called an active redundant configuration. The parallel system may occur as a result of the basic system structure or may be produced as a results of additional spares included to improve the system reliability. The parallel system configuration is successful if any one of the m subsystems is successful. In other words, the system fails when all m subsystems fail. Thus the unreliability of the parallel system, when all subsystems are independent, is the product of the unreliabilities of its components

$$Q_{\mathrm{s}} = \prod_{i=1}^{m} Q_i \,,$$

$$R_{\mathrm{s}} = 1 - Q_{\mathrm{s}} = 1 - \prod_{i=1}^{m} Q_i = 1 - \prod_{i=1}^{m} (1 - R_i) \,. \tag{54.2}$$

Standby Configuration

Standby components are used to increase the reliability of the system. There are three types of redundancy [54.6, 7]: (1) cold standby, (2) warm standby, and (3) hot

standby. In the cold-standby redundancy configuration, a component (or subsystem) does not fail before it is put into operation. In warm-standby redundancy configuration, the component can fail in standby mode. However, the chances of failure (failure rate in standby) are less than the chances of failure in operation (failure rate in operation). If the failure pattern of a standby component does not depend on whether the component is idle or in operation, then it is called hot standby. In this chapter, the analysis is provided for cold- and hot-standby components only.

If the redundant components operate simultaneously from time zero, even though the system needs only one of them at any time, such an arrangement is called parallel (or active) redundancy. This arrangement is essential when switching or starting a good component following a component failure is ruled out. The mathematical models for hot-standby and parallel redundancy arrangements are equivalent. In the cold-standby redundancy configuration, the redundant components are sequentially used in the system at component failure times. The system fails when all components fails. Cold-standby redundancy provides longer system life than hot-standby redundancy.

For a cold-standby system with a total of m components where initially the first component is in operation and the rest of the $m-1$ components are in standby, the system reliability at time t is

$$\begin{aligned} &R_\mathrm{s}(t) \\ &= \Pr(\text{system operates successfully until time } t) \\ &= \Pr(X_1 + X_2 + \cdots + X_\mathrm{m} \geq t)\,, \end{aligned} \tag{54.3}$$

where X_i is the random variable that represents the failure time of component i. The final expression for the system reliability is a function of the parameters of the component failure-time distribution. If all components are identical and follow exponential failure-time distributions with hazard rate λ, then the reliability of the cold-standby system with m components is [54.7, 8]

$$\begin{aligned} R_\mathrm{s}(t) &= \exp(-\lambda t)\left(\sum_{i=0}^{m-1}\frac{(\lambda t)^i}{i!}\right) \\ &= p\left(\sum_{i=0}^{m-1}\frac{[-\ln(p)]^i}{i!}\right)\,, \end{aligned} \tag{54.4}$$

where $p = \exp(-\lambda t)$ is the reliability of each component.

k-out-of-n Active Standby Configuration

In this configuration, the system consists of n active components in parallel. The system functions successfully when at least k of its n components function. This is also referred to as a k-out-of-n:G parallel (or hot or active standby) system or simply a k-out-of-n system. A parallel system is a special case of a k-out-of-n system with $k = 1$, i. e., a parallel system is a 1-out-of-n system. Similarly, a series system is a special case of a k-out-of-n system with $n = 1$, i. e., a series system is an n-out-of-n system.

When all components (or subsystems) are independent and identical, the reliability of a k-out-of-n system with component reliability equal to p is [54.3, 8, 9]:

$$\begin{aligned} R_\mathrm{s} &= \sum_{i=k}^{n}\binom{n}{i}p^i(1-p)^{n-i} \\ &= 1 - \sum_{i=0}^{k-1}\binom{n}{i}p^i(1-p)^{n-i} \\ &= \mathrm{binfc}(k-1;\, p, n)\,. \end{aligned} \tag{54.5}$$

k-out-of-n Cold-Standby Configuration

In this configuration, the system consists of a total of n components. Initially, only k components are in operation and the remaining (n-k) redundant components are kept in cold standby. The redundant components are sequentially used to replace the failed components. When all components (or subsystems) are identical and the failure distribution is exponential, the reliability of a k-out-of-n cold-standby system is [54.8]:

$$\begin{aligned} R_\mathrm{s}(t) &= \exp(-k\lambda t)\left(\sum_{i=0}^{k-1}\frac{(k\lambda t)^i}{i!}\right) \\ &= p^k\left(\sum_{i=0}^{k-1}\frac{[-k\ln(p)]^i}{i!}\right)\,. \end{aligned} \tag{54.6}$$

General System Configuration

It is well known that the reliability of a system is a function of the component reliabilities or its parameters. Hence, we have [54.8]

$$R_\mathrm{s} = h(R_1, \cdots, R_\mathrm{m})\,. \tag{54.7}$$

Here, h is the function that represents the system reliability in terms of the component reliabilities. The actual formula for $h(R_1, \cdots, R_\mathrm{m})$ depends on the system configuration. Several algorithms are available [54.7] to compute (54.7). Recent literature [54.10] has shown

that, in many cases, algorithms based on binary decision diagrams are more efficient in computing (54.7).

54.1.5 System Availability

If the objective is to maximize the system availability, then we should express the system availability in a mathematical form. The form of the availability expression varies with system configuration. In this section, we provide the availability expressions for some specific configurations. These expressions are valid under the following assumptions.

1. The failure- and repair-time distributions of all components are independent.
2. There are sufficient repair resources such that repair of a failed component starts immediately.

Series Configuration

The availability of the series system is [54.7]

$$A_s = \prod_{i=1}^{m} A_i \,, \tag{54.8}$$

where A_i is the availability of component (subsystem) i and m is the number of components.

Parallel Configuration

The availability of the parallel system is [54.7]:

$$A_s = 1 - \prod_{i=1}^{m} A_i = 1 - \prod_{i=1}^{m} (1 - A_i) \,, \tag{54.9}$$

where A_i is the availability of component (subsystem) i and m is the number of components.

k-out-of-n Active-Standby Configuration

The availability of a k-out-of-n system with identical components is [54.7, 9]

$$\begin{aligned} A_s &= \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \\ &= \text{binfc}(k-1;\, p, n) \,, \end{aligned} \tag{54.10}$$

where p represents the component availability.

General System Configuration

When failure and repair processes of all components or subsystems are independent, then we can represent the system availability as a function of component availabilities. Hence, we have

$$R_s = h(A_1, \cdots, A_m). \tag{54.11}$$

Here, h is the function that represents the system availability in terms of component availabilities. The actual formula for $h(A_1, \cdots, A_m)$ depends on the system configuration. In this cases, the same algorithms used for system reliability can be used for computing system availability.

54.1.6 Other Objective Functions

Similarly, we can also find the other objective functions analytically [54.11–17]. When closed-form expression are not available the objective function can be calculated using either numerical methods or simulation [54.7]. In such cases, the methods for finding the optimal solutions should not depend on the form of the objective function. The generic nature of these solution methods may lose the advantages associated with the specific form of the objective function. In this section, we provide the expressions for some of the objective functions.

Unreliability

In some systems, the objective would be minimization of unreliability. The unreliability of a system can be obtained from its reliability [54.7]

$$\begin{aligned} &\text{Unreliability} = 1 - \text{Reliability}, \\ &Q_s = 1 - R_s \,. \end{aligned} \tag{54.12}$$

Unavailability

In some systems, the objective would be minimization of unavailability. We have [54.7]

$$\begin{aligned} &\text{Unavailability} = 1 - \text{Availability}, \\ &U_s = 1 - A_s \,. \end{aligned} \tag{54.13}$$

Total Uptime

In some systems, the objective would be the maximization of total uptime (or operational time) over a period of time. We have [54.7]

$$TUT_s(t) = \int_0^t A_s(x)\,\mathrm{d}x \,. \tag{54.14}$$

Total Downtime

In some systems, the objective would be minimization of total downtime during a specified period of time. We have [54.7]

$$TDT_s(t) = \int_0^t U_s(x)\,\mathrm{d}x = t - TUT_s(t) \,. \tag{54.15}$$

Mean Availability

In some systems, the objective would be maximization of system mean availability over a period of time. We have [54.7]

$$A_{\mathrm{M}}(t) = \frac{TUT_{\mathrm{s}}(t)}{t} \,. \tag{54.16}$$

Mean Unavailability

In some systems, the objective would be minimization of mean unavailability over a period of time. We have [54.7]

$$U_{\mathrm{M}}(t) = \frac{TDT_{\mathrm{s}}(t)}{t} = 1 - A_{\mathrm{M}}(t) \,. \tag{54.17}$$

Average cost

In the majority of applications, the objective of system design is to minimize the overall cost associated with the system. The total cost is the sum of several cost factors. These include:

1. System failure cost,
2. Cost of components and spares,
3. Cost of maintenance (repair, replacement, and inspection costs),
4. Warranty costs,
5. Cost associated with downtime (or loss of production).

The actual formulas and cost factors vary with the applications. For details, see [54.11–15, 17].

Decision Variables

These are the values that we are interested to find such that the specified objective is minimized or maximized [54.1]. The decision variables include:

1. Type of system configuration,
2. Types of components or spares,
3. Number of spares in a specific application (or subsystem),
4. Number of repair personnel.

The type of a component is applicable if there are several alternative components for the same application with various costs, weights, volumes, and failure rates (or reliabilities).

Constraints

The optimal solutions should be obtained within the allowed resource restrictions. These are also called constraints of the optimization problem. The constraints include:

1. Desired reliability,
2. Desired availability,
3. Desired mean time to failure (MTTF) or mean time between failures (MTBF),
4. Allowed downtime,
5. Allowed unavailability,
6. Allowed budget (for spares or repair resources),
7. Allowed weight,
8. Available space (or volume).

54.1.7 Existing Optimization Models

As described in the previous sections, there are several possibilities for objective functions, constraints, and decision variables. Furthermore, the diversity of the system configurations and their special properties lead to several optimization models. In this section, we present some well-studied optimization models and associated mathematical formulations. A detailed treatment of these problems is presented in [54.6].

In all these models, it is assumed that the system consists of several stages (subsystems or modules). In most cases, the objective is to maximize the system reliability by optimally assigning the component reliabilities and/or redundancy levels at various stages, subject to resources constraints.

Allocation of Continuous Component Reliabilities

In this formulation, the system reliability can be improved through the selection of component reliabilities at stages subject to resource constraints. Therefore, the decision variables are the reliabilities of the stages $(r_1, \cdots, r_m)$. The problem of maximizing system reliability through the selection of component reliabilities subject to resource constraints can be expressed as:

$$\begin{aligned} &\text{Maximize } R_{\mathrm{s}} = f(r_1, \cdots, r_m) \\ &\quad \text{subject to:} \\ &g_i(r_1, \cdots, r_m) \le b_i \qquad \text{for } i = 1, \cdots, k \,, \\ &r_j^{\mathrm{l}} \le r_j \le r_j^{\mathrm{u}} \qquad \text{for } j = 1, \cdots, m \,. \end{aligned} \tag{54.18}$$

This is a nonlinear programming problem.

Allocation of Discrete and Continuous Component Reliabilities

In this formulation, we have u_j discrete choices for component reliability at stage j for $j = 1, \cdots, s$ and the choice for the component reliability at stages $s+1, \cdots, m$ is on a continuous scale. Therefore, the decision variables are the reliabilities of stages $(r_1, \cdots, r_m)$. Because we have discrete choices for the reliability

at stage j for $j = 1, \cdots, s$, selecting the reliability at each stage is equivalent to selecting the related choice, which can be expressed as an integer. Hence, the decision variables are: $(x_1, \cdots, x_s, R_{s+1}, \cdots, R_m)$. Alternatively, they are equivalent to specifying $[R_1(x_1), \cdots, R_s(x_s), R_{s+1}, \cdots, R_m]$. The problem of maximizing system reliability through the selection of component reliabilities subject to resource constraints can be expressed as:

$$\begin{aligned}
&\text{Maximize} \\
&R_s = f(R_1(x_1), \cdots, R_s(x_s), R_{s+1}, \cdots, R_m) \\
&\text{subject to:} \\
&g_i[R_1(x_1), \cdots, R_s(x_s), R_{s+1}, \cdots, R_m] \le b_i \\
&\qquad\qquad \text{for } i = 1, \cdots, k\,, \\
&x_j \in \{1, 2, \cdots, u_j\} \qquad \text{for } j = 1, \cdots, s\,, \\
&r_j^{\mathrm{l}} \le r_j \le r_j^{\mathrm{u}} \qquad \text{for } j = 1, \cdots, m\,.
\end{aligned} \tag{54.19}$$

This problem is called the reliability allocation problem. This problem can be simplified when the separability assumption is applicable. With the separability assumption, we have

$$g_i(R_1, \cdots, R_m) = \sum_{j=1}^{m} g_{ij}(r_i)\,. \tag{54.20}$$

This problem is a nonlinear mixed integer programming problem.

Redundancy Allocation

System reliability can be improved through the selection of redundancy levels at stages, subject to resource constraints. Therefore, the decision variables are the redundancy levels of stages $(x_1, \cdots, x_m)$. Alternatively, they are equivalent to specifying $[R_1(x_1), \cdots, R_m(x_m)]$. The problem of maximizing the system reliability through the selection of optimal redundancy levels, $x_1, \cdots, x_n$, subject to resource constraints can be expressed as:

$$\begin{aligned}
&\text{Maximize } R_s = f(x_1, \cdots, x_n) \\
&\text{subject to:} \\
&g_i(x_1, \cdots, x_n) \le b_i \qquad \text{for } i = 1, \cdots, k\,, \\
&l_j \le x_j \le u_j \qquad \text{for } j = 1, \cdots, m\,, \\
&x_j \qquad \text{is an integer}\,.
\end{aligned} \tag{54.21}$$

This problem is called the redundancy allocation problem. It is a nonlinear integer programming problem. As in other cases, this problem can be simplified with the separability assumption, which is often applicable in real life. With the separability assumption, we have

$$g_i(x_1, \cdots, x_m) = \sum_{j=1}^{m} g_{ij}(x_i)\,. \tag{54.22}$$

This problem is thoroughly discussed in the literature [54.3, 4, 6, 18–21].

Reliability–Redundancy Allocation

In this formulation, the system reliability can be improved through the selection of component reliabilities as well as redundancy levels at stages, subject to resource constraints. Therefore, the decision variables are the pair of values containing the reliability and level of redundancy for each stage $s(r_s, x_s)$. Hence, the decision variables are both $(x_1, \cdots, x_n)$ and $(r_1, \cdots, r_n)$. The problem of finding simultaneously the optimal redundancy levels $(x_1, \cdots, x_n)$ and the optimal component reliabilities $(r_1, \cdots, r_n)$ that maximize system reliability subject to the resource constraints can be expressed as:

$$\begin{aligned}
&\text{Maximize} \\
&R_s = f(x_1, \cdots, x_n; r_1, \cdots, r_n) \\
&\text{subject to:} \\
&g_i(x_1, \cdots, x_n; r_1, \cdots, r_n) \le b_i \quad \text{for } i = 1, \cdots, k\,, \\
&l_j \le x_j \le u_j \qquad \text{for } j = 1, \cdots, m\,, \\
&r_j^{\mathrm{l}} \le r_j \le r_j^{\mathrm{u}} \qquad \text{for } j = 1, \cdots, m\,, \\
&x_j \qquad \text{is an integer}\,.
\end{aligned} \tag{54.23}$$

This problem is called the reliability-redundancy allocation problem. It is a nonlinear mixed integer programming problem. This problem can also be simplified when the separability assumption is applicable. With the separability assumption, we have:

$$g_i(x_1, \cdots, x_m; r_1, \cdots, r_n) = \sum_{j=1}^{m} g_{ij}(x_i, r_i)\,. \tag{54.24}$$

Redundancy Allocation for Cost Minimization

In this formulation, the objective is cost minimization. In traditional models, the overall cost of the system is expressed as the sum of the cost of all components (stages). The cost of each stage is a function of the redundancy level of the stage. Therefore, the decision variables are the redundancy levels at states $(x_1, \cdots, x_n)$. The problem of finding simultaneously optimal the redundancy levels $(x_1, \cdots, x_n)$ that minimize the system cost subject

to resource constraints can be expressed as:

$$\begin{aligned} &\text{Minimize } C_s = \sum_{j=1}^{m} C_j(x_j) \\ &\text{subject to:} \\ &g_i(x_1, \cdots, x_n) \le b_i \qquad \text{for } i = 1, \cdots, k\,, \\ &l_j \le x_j \le u_j \qquad \text{for } j = 1, \cdots, m\,, \\ &x_j \qquad \text{is an integer}\,. \end{aligned} \tag{54.25}$$

Here, $c_j(x_j)$ is the cost of x_j components at stage j.

Cost-minimization problems are less studied in the literature. Furthermore, there is a lot of scope to improve this formulation by incorporating various cost factors associated with system.

Other Formulations

Other formulations include:

1. Allocation of discrete component reliabilities and redundancies. In this formulation, depending on the type of stage, we can select either a discrete component reliability or the redundancy level (which is also discrete). Therefore, it is a nonlinear integer programming problem. A generalization to this formulation could be by allowing a combination of discrete and continuous choices for component reliability along with choosing the redundancies levels at each stage. Hence, this generalization problem becomes a mixed integer nonlinear programming problem.
2. Component assignment problem. This is applicable when components at various stages can be interchangeable
3. Multi-objective optimization problem. This is applicable when there are multiple simultaneous objectives such as maximization of reliability and minimization of cost, volume, weight, etc.

For more details on these formulations, see [54.6].

54.2 Cost-Effective Designs

In most of the problems studied in the literature, the objective is to maximize the system reliability. However, as we discussed earlier, the optimal solution that maximizes reliability may not necessarily minimizes the overall cost associated with the system. Even though some formulations allow the specification of cost factors in the constraints, they do not minimize the cost. Instead they provide optimal solutions within the specified cost constraints, such as an allowed budget. Further, the well-known cost-minimization problem minimizes the total cost of components subject to other constraints such as desired reliability and allowed volume, weight, etc. However, the cost of the system not only includes the cost of components but also includes various other factors, which are discussed in Sect. 54.1.6. The optimal solution should strike a balance between various competing cost factors such that the overall cost of the system is minimized and at the same time all constraints are satisfied. The cost factors varies with the system type, failure mode, and other details related to a specific system.

54.2.1 Nonrepairable Systems

General Cost Structure

For nonrepairable systems, the major cost factors are the cost of spares and the cost of failure. The system failure cost can be minimized by using reliable system designs. System reliability can be improved by using the redundancy of spares, which can be kept in hot- or cold-standby mode (or in some cases in warm-standby mode). Although the reliability of a system increases with the addition of spares, the cost of the system also increases due to the number of redundant components added to the system. Thus, it is desirable to derive a cost-effective solution that strikes a balance between the system failure cost and the cost of spares in the system. Therefore, the objective is to minimize the average cost associated with the system

$$\begin{aligned} &\text{Average cost} \\ &= \text{Cost of minimum required components} \\ &\quad + \text{Cost of spares} + \text{Cost of failure}\,. \end{aligned} \tag{54.26}$$

Because the cost of the minimum required components is a fixed initial cost, this cost can be eliminated. Alternatively, minimizing the cost using (54.26) and (54.27) are mathematically the same and produce the same optimal solution for the spares. Therefore, depending on convenience, we can use one of these formulations

$$\text{Average cost} = \text{Cost of spares} + \text{Cost of failure}\,. \tag{54.27}$$

The cost of failure can be modeled in several ways. There can be a cost (fixed or randomly distributed with a finite mean) for not completing the mission successfully.

Therefore,

$$\begin{aligned}&\text{Average cost of failure}\\&= \text{Cost of failure of a mission} \times \text{Unreliability}\,.\end{aligned} \tag{54.28}$$

For example, this kind of cost is applicable for missions such as landing on a planet or a moon. Finally, the average system cost is

$$\begin{aligned}\text{Average cost} &= \text{Cost of spares}\\&\quad + \text{Cost of failure of a mission}\\&\quad \times \text{Unreliability}\,.\end{aligned} \tag{54.29}$$

In some scenarios the cost of system failure may depend on the time at which the failure occurs. If the system fails at the beginning of the mission, the losses are high compared to if the system fails almost at the end of the mission. If the cost is linearly proportional to the remaining mission time, then we can express the average system cost in terms of the average remaining mission time. Therefore,

$$\begin{aligned}&\text{Average cost of failure}\\&= \text{Cost of failure of a mission per unit time}\\&\quad \times \text{Remaining mission}\\&= \text{Cost of failure of a mission per unit time}\\&\quad \times \text{Mission duration}\\&\quad \times \text{Average unreliability or unavailability}\\&= \text{Cost of failure of a mission per unit time}\\&\quad \times \text{Mean downtime}\,.\end{aligned} \tag{54.30}$$

In most cases, the problem of minimizing the average cost of the system can be converted into the problem of maximizing the average profit of the system. When there is a fixed profit for each successful operation of the mission, we can easily show that these two problems are identical.

$$\begin{aligned}&\text{Average system profit}\\&= \text{Profit during success} \times \text{Reliability}\\&\quad - \text{Losses during failure} \times \text{Unreliability}\\&\quad - \text{Cost of components}\\&= \text{Profit during success}\\&\quad - (\text{Profit during success} - \text{Losses during failure})\\&\quad \times \text{Unreliability}\\&\quad - \text{Cost of components}\,.\end{aligned} \tag{54.31}$$

Because the *profit during success* is fixed, it is a constant. Therefore, the problem of maximizing the system profit is reduced to maximization of Function1 shown in (54.32):

$$\begin{aligned}&\text{Function1}\\&= -(\text{Profit during success} - \text{Losses during failure})\\&\quad \times \text{Unreliability}\\&\quad - \text{Cost of components}\,.\end{aligned} \tag{54.32}$$

Because maximizing $[f(x)]$ is equivalent to minimizing $[-f(x)]$, the problem is equivalent to minimizing the Function2 in (54.33). For details, see [54.1].

$$\begin{aligned}&\text{Function2}\\&= (\text{Profit when success} - \text{Losses during failure})\\&\quad \times \text{Unreliability}\\&\quad + \text{Cost of components}\,.\end{aligned} \tag{54.33}$$

The objective functions shown in (54.29) and (54.33) have the same form. Therefore, the same methods that are used for cost-minimization problems can also be used for profit-maximization problems.

A specific Model for Nonrepairable Systems

In this section, we provide a detailed model for a specific class of nonrepairable systems. The system consists of several subsystems, which are arranged in a network configuration. Each subsystem requires a minimum number of identical components to perform its functions. The additional spare components in each subsystem can be kept online or in cold- or hot-standby mode. Therefore, each subsystem behaves like a k-out-of-n:G cold/hot-standby system. The objective of the problem considered in this chapter is to find the optimal cost-effective design of the overall system. This means that the optimal number of components in each subsystem must be found to minimize the overall cost associated with the system.

Assumptions.

1. The system consists of m subsystems. The subsystems can be arranged in a complex network configuration [54.22].
2. System structure function is coherent with respect to the subsystems.
3. All subsystems are statistically independent.
4. Each subsystem consists of $n_i \equiv k_i + s_i$ identical components.
5. Subsystem i requires k_i components for its successful operation.
6. Subsystem i can have s_i spares. Spares of a subsystem can be kept in hot- or cold-standby mode.

7. The failure rate of an operational component is constant. The failure rate of a hot-standby component is equivalent to the failure rate of an operational component. The failure rate of a cold-standby component is zero.

Problem Formulation. Because the subsystems are independent, we can compute the reliability of each subsystem separately and use those results to compute the overall system reliability. The reliability calculation of a subsystem depends on the type of spares used.

1. Hot standby

$$R_i = \text{binfc}(k_i - 1;\, p_i, n_i)\,. \tag{54.34}$$

2. Cold standby

$$\begin{aligned} R_i &= \exp(-k_i\lambda_i t)\left(\sum_{j=0}^{s_i}\frac{(k_i\lambda_i t)^j}{j!}\right) \\ &= p_i^{k_i}\left(\sum_{j=0}^{s_i}\frac{[-k_i\ln(p_i)]^j}{j!}\right)\,. \end{aligned} \tag{54.35}$$

System reliability is a function of the subsystem reliabilities

$$\begin{aligned} R(\boldsymbol{n}) &= h(R_1, \cdots, R_m) \\ &= R_i h_i^1 + (1 - R_i)h_i^0 \\ &= \left(h_i^1 - h_i^0\right)R_i + h_i^0\,. \end{aligned} \tag{54.36}$$

The average cost of the system, $T_s(n)$, is the cost incurred when the system has failed, plus the cost of all components in the system

$$T(\boldsymbol{n}) = \sum_{i=1}^{m} c_i n_i + c_f\left[1 - R(\boldsymbol{n})\right]\,. \tag{54.37}$$

The objective is to find the optimal $\boldsymbol{n}$ that minimizes $T(\boldsymbol{n})$. The problem can be further refined by considering minimum acceptable reliability (R_a), acceptable upper limit on total weight & volume, and allowable budget for spares acquisition. Therefore, the constraints are

$$\begin{aligned} &\text{Budget constraint:} && \sum_{i=1}^{m} s_i.c_i \le B \\ &\text{Volume constraint:} && \sum_{i=1}^{m} v_i \le V \\ &\text{Weight constraint:} && \sum_{i=1}^{m} w_i \le W \\ &\text{Reliability constraint:} && R(\boldsymbol{n}) \ge R_a\,, \end{aligned} \tag{54.38}$$

where B is the allowed budget, V is the maximum allowed volume, W is the maximum allowed weight, and R_a is the minimum acceptable reliability.

In addition, we can also add explicit constraints on the number of components in each subsystem. However, in general, the problem is difficult to solve when no explicit constraints on the decision variables are specified. The solution can be simplified by finding the explicit bounds for each decision variable, i. e., the number of components. It is a nonlinear integer programming problem.

54.2.2 Repairable Systems

General Cost Structure

For repairable systems, the major cost factors are:

- initial cost of spares,
- cost of failures,
- cost of repair and replacement,
- cost of storage.

However, in the long run, the cost of the initial cost of spares is negligible compared to the other costs. The cost of failure can be minimized by minimizing the system unreliability, which in turn can be achieved by increasing the number of spares in the system. However, the increase in the number of spares can also increase the maintenance and operational costs of spares, which include repair, replacement and storage costs. Thus, it is desirable to derive a cost-effective solution that strikes a balance between the system failure cost and the cost of maintenance and operation

$$\begin{aligned} &\text{Average system cost} \\ &= \text{Cost of maintenance} \\ &\quad + \text{Cost of failure} \times \text{Unavailability}\,. \end{aligned} \tag{54.39}$$

However, for short-duration systems, we have

$$\begin{aligned} &\text{Average system cost} \\ &= \text{Initial cost of components and spares} \\ &\quad + \text{Cost of set up} \\ &\quad + \text{Cost of maintenance} \\ &\quad + \text{Cost of failure} \times \text{Unavailability} \\ &\quad + \text{Cost of disposal}\,. \end{aligned} \tag{54.40}$$

The cost of disposal can be positive or negative (for resale value it is negative). When we can calculate the cost of each component including the cost of procurement,

set up, and resale values. The problem can be reduced to

$$\begin{aligned}&\text{Average system cost}\\&= \text{Effective cost of components and spares}\\&\quad + \text{Cost of maintenance}\\&\quad + \text{Cost of failure} \times \text{Unavailability}\,.\end{aligned} \tag{54.41}$$

A Specific Model for Repairable Systems

The majority of engineering and manufacturing systems consist of several subsystems [54.22], which are usually nonidentical. In general, all subsystems need not be in series, but can be arranged in any configuration. The success logic of these types of systems can be represented using networks or reliability block diagrams, which are collectively known as combinatorial reliability models [54.23]. Each subsystem can have one or more functionally similar components [54.24]. For successful operation of a subsystem, there must be at least a specified number of components in operation. Such subsystems, known as k-out-of-n subsystems [54.9], have a wide range of applications [54.9, 12, 13, 25]. A special case of a k-out-of-n subsystem with $k = 1$ is called a parallel subsystem [54.26].

In general, systems and components are repairable. Therefore, systems and components undergo several failure–repair cycles that include logistic delays while performing repairs [54.27]. In the long run, the downtime costs are directly related to the asymptotic unavailability of the system. System unavailability can be reduced by increasing the availability of its subsystems, which in turn can be increased by additional spares for each subsystem. Although the availability of a system increases with the addition of spares, the cost of the system also increases due to the added operational and maintenance costs. Thus, it is desirable to derive a cost-effective solution that strikes a balance between the system downtime cost and the operational and maintenance costs of spares in the system. The objective of the cost-effective solution is to find the optimal number of spares in each subsystem that minimizes the overall system cost.

Assumptions.

1. The system consists of m subsystems. All subsystems are statistically independent.
2. Subsystem i consists of $n_i \equiv k_i + s_i$ components, where subsystem i requires at least k_i components for its successful operation.
3. Failure, repair, and logistic delay times of all components are independent and can follow any distribution.
4. System failure cost is proportional to the downtime.
5. There is a fixed miscellaneous cost for each repair of a component. In addition to the fixed cost, the cost of repair has a variable cost, which is proportional to the amount of repair time. The downtime of a component is the sum of the repair time and logistic delay time.

Problem Formulation.

The average system cost is the sum of the average cost of downtime plus the average cost of maintenance

$$T(\boldsymbol{n}) = T_{\mathrm{d}}(\boldsymbol{n}) + T_{\mathrm{m}}(\boldsymbol{n})\,. \tag{54.42}$$

The cost of downtime can be calculated from the percentage of downtime within a unit time duration and the loss (cost) per unit downtime. As time progresses, the system reaches a steady-state condition. Under steady-state conditions, the percentage of downtime is equivalent to the steady-state unavailability. Hence,

$$T_{\mathrm{d}}(\boldsymbol{n}) = c_{\mathrm{f}} U(\boldsymbol{n}) = c_{\mathrm{f}}\left[1 - A(\boldsymbol{n})\right]\,. \tag{54.43}$$

System availability is a function of the subsystem availabilities

$$A(\boldsymbol{n}) = h(A_1, \cdots, A_{\mathrm{m}})\,. \tag{54.44}$$

The actual formula for $h(A_1, \cdots, A_m)$ depends on the system configuration. For example, if the system configuration is series, then

$$A(\boldsymbol{n}) = \prod_{i=1}^{m} A_i(n_i)\,. \tag{54.45}$$

Several algorithms are available [54.7] to compute (54.44). Recent literature [54.10] has shown that, in many cases, algorithms based on binary decision diagram are more efficient in computing (54.44).

If the system is under steady-state conditions, and the failure and repair processes of all components are independent, the availability of each component can be calculated using its mean time to failure (MTTF), mean time to repair (MTTR), and mean logistic delay time (MLDT). Because each subsystem consists of identical components and the configuration of each subsystem is k_i-out-of-n_i, we have:

$$\begin{aligned}A_i = A_i(n_i) &= \mathrm{binf}(n_i - k_i;\, q_i, n_i)\\&= \mathrm{binfc}(k_i - 1;\, p_i, n_i)\end{aligned} \tag{54.46}$$

where p_i and q_i are, respectively, the operational availability and unavailability of a component in subsystem i. Furthermore, given that $q_i = 1 - p_i = (\psi_i + \varphi_i)/(\phi_i$

$+\psi_i+\varphi_i)$, where ϕ_i is the MTTF, ψ_i is the MTTR, and φ_i is the MLDT of a component in subsystem i.

The cost of maintenance is proportional to the cost associated with the repairs of individual components. The cost of repair of a failed component includes the miscellaneous fixed cost as well as the variable cost based on the repair time. Therefore, if the repair of a failed component in the subsystem i takes on average ψ_i units of time, then the average cost of repair for each instance of repair is

$$R_i = \gamma_i + \psi_i \delta_i \; . \tag{54.47}$$

Let ν_i be the failure frequency, i. e., expected number of failures per unit time, of a component in subsystem i. Because all components in a subsystem are identical, on average, we have $N_i \equiv n_i f_i$ failures per unit time. Under steady-state conditions, we have

$$\begin{aligned} \nu_i &= \frac{1}{\phi_i+\psi_i+\varphi_i} \\ N_i &= \frac{n_i}{\phi_i+\psi_i+\varphi_i} \; . \end{aligned} \tag{54.48}$$

Hence, the cost of maintenance of a subsystem per unit time is $\theta_i(n_i)$

$$\theta_i(n_i) = N_i R_i = n_i \nu_i R_i \; . \tag{54.49}$$

The cost of maintenance of the entire system is

$$T_{\mathrm{m}}(\boldsymbol{n}) = \sum_{i=1}^{m} \theta_i(n_i) = \sum_{i=1}^{m} n_i \nu_i R_i \; . \tag{54.50}$$

It should be noted that, in the long run, the initial cost of the spares per unit time is negligible and need not be considered.

Therefore, the average cost of the system is

$$\begin{aligned} T(\boldsymbol{n}) &= \sum_{i=1}^{m} c_i n_i + c_{\mathrm{f}} U(\boldsymbol{n}) \; , \\ c_i &= \frac{\gamma_i + \psi_i \delta_i}{\phi_i+\psi_i+\varphi_i} \; , \\ U(\boldsymbol{n}) &= 1 - A(\boldsymbol{n}) \; , \\ A(\boldsymbol{n}) &= h(A_1, \cdots, A_m) \; , \\ A_i &\equiv A_i(n_i) = \mathrm{binf}(n_i - k_i; q_i, n_i) \; , \\ q_i &= \frac{\psi_i+\varphi_i}{\phi_i+\psi_i+\varphi_i} = 1 - \frac{\phi_i}{\phi_i+\psi_i+\varphi_i} \; . \end{aligned} \tag{54.51}$$

The objective is to find the optimal $\boldsymbol{n}$ that minimizes $T(\boldsymbol{n})$. The problem can be further refined by considering the maximum acceptable unavailability (U_{a}) and the acceptable upper limit on total weight and volume. Therefore, the constraints are

$$\begin{aligned} \text{Volume constraint:} \quad & \sum_{i=1}^{m} v_i \le V \; ; \\ \text{Weight constraint:} \quad & \sum_{i=1}^{m} w_i \le W \; ; \\ \text{Unavailability constraint:} \quad & U(\boldsymbol{n}) \le U_{\mathrm{a}} \; . \end{aligned} \tag{54.52}$$

Similarly, we can also add explicit constraints on the number of components in each subsystem. This is a nonlinear integer programming problem. More specific forms of average cost functions are studied in [54.11–15, 17].

54.3 Optimal Design Algorithms

54.3.1 An Overview

In this section, we discuss various methods for solving optimal system problems. We demonstrate these methods for solving the problems associated with cost-effective designs. The same algorithms can also be applied for the other problem formulations. The algorithms can be broadly classified as follows:

1. *Exact methods*. These methods produce exact optimal solutions. It is generally difficult to develop efficient exact methods for spares optimization problems. This is because the objective function, which is the function to be minimized, is nonlinear and involves several integer variables, which are the components to be optimized. In addition, the objective function may have several peaks and may not possess monotonic increasing or decreasing characteristics. Therefore, exact methods involve more computational effort and usually require large amounts of computer memory.
   One exact method is exhaustive searching, where the objective function is computed for all possible combinations of the decision variables, which would be the quantities of spares. However, exhaustive searching is infeasible even for a moderately size problem for optimizing the number of spares. Other algorithms in this category involve dynamic

programming, branch-and-bound methods, and nonlinear mixed integer programming techniques. Some researchers have developed exact methods that have closed-form or computationally efficient solutions for some well-structured systems such as *k*-out-of-*n* systems, parallel–series, and series–parallel systems.

2. *Approximation methods*. Some of the difficulties of spares optimization problem are due to the presence of an integer variable (the number of spares), the variable forms of cost functions (the number of terms in the cost function is based on the number of spares), and non-polynomial cost functions. In addition, the cost function may have several peaks and valleys. By using approximation methods, these complex problems can be modeled in a simpler form by considering continuous variables for spares quantities and approximate forms for the cost function. This approach produces near-optimal solutions, or solutions that are very close to the exact solution result. However, approximation methods cannot guarantee the global optimal solutions, which are exact solutions that actually minimize the objective functions.
3. *Heuristic methods*. Finding exact optimal solutions for large complex systems is not always feasible because such problems involve resource-intensive computation efforts and usually require large amounts of computer memory. For these reasons, researchers in reliability optimization have placed more emphasis on heuristic approaches, which are methods based on rules of thumb or guidelines that generally find the solutions but do not guarantee an exact solution. In most of these approaches, the solution is improved in each iteration. One of the simple heuristic approaches is the greedy method, where the quantity of spares is incremented for the subsystem where the maximum reduction in cost is achieved. This iterative process is stopped when a point is reached where adding spares to any component increases the cost. Heuristic methods typically produce local optimal solutions within a short time. However, they may not guarantee the global optimal solutions and may produce local optimal solutions, which is an optimal solution in a local neighborhood. For spares optimization, there may exist several local optimal solutions, where changing any variable slightly (increasing or decreasing any spare) increases the total cost. The global optimal solution corresponds to the minimal solution out of all such local optimal solutions.
4. *Meta-heuristic methods*. These methods are based more on artificial reasoning than classical mathematics-based optimization. They include genetic algorithms (GA) and simulated annealing (SA). GAs seek to imitate the biological phenomenon of evolutionary production through parent–child relationships. SA is based on a physical process in metallurgy. Most of these methods use stochastic searching mechanisms. Meta-heuristic methods can overcome the local optimal solutions and, in most cases, they produce efficient results. However, they also cannot guarantee the global optimal solutions.
5. *Hybrid methods*. These methods use combinations of different optimization techniques. An example of a hybrid method would be to find the initial solution with different heuristic approaches or approximations and then apply meta-heuristic methods to search for better solutions.

54.3.2 Exact Methods

In order to find an exact solution to the optimization problem, we should either compute the objective function for all possible combinations of decision variables or systematically identify all potential combinations of decision variables. Therefore, exact methods are appli-

Table 54.1 Exhaustive search results

n_1	n_2	R_s	C_s	Remarks
1	1	0.600 000	–	C5 violated
1	2	0.720 000	–	C5 violated
1	3	0.744 000	–	C5 violated
1	4	0.748 800	–	C5 violated
2	1	0.750 000	–	C5 violated
2	2	0.900 000	106.000 000	
2	3	0.930 000	78.000 000	
2	4	0.936 000	74.000 000	
3	1	0.787 500	–	C5 violated
3	2	0.945 000	62.000 000	
3	3	0.976 500	32.500 000	
3	4	0.982 800	28.200 000	
4	1	0.796 875	–	C5 violated
4	2	0.956 250	51.750 000	
4	3	0.988 125	21.875 000	
4	4	0.994 500	17.500 000	
5	1	0.799 219	–	C5 violated
5	2	0.959 062	49.937 500	
5	3	0.991 031	19.968 750	
5	4	0.997 425	–	C3, C4 violated

cable only when the problem size is very small or it posses a special properties that can be used to identify the potential optimal solutions. Many exact methods were developed before 1980 and documented in *Tillman* et al. [54.4].

Exhaustive Searching

In this method, compute the objective function for all possible combination of decision variable. In order to achieve this goal, we should know the lower and upper limits on the decision variables. This method is demonstrated for a nonrepairable system consisting of two subsystems in series. The objective is to find the optimal number of online spares in each subsystem that minimizes the average cost associated with the system

$$\begin{aligned}
&\text{Minimize } C_s = n_1c_1 + n_2c_2 + c_f[1 - R_s(n_1, n_2)] \\
&\quad \text{where,} \qquad (54.53)\\
&R_s(n_1, n_2) = R_1(n_1)R_2(n_2)\,, \\
&R_1(n_1) = 1 - (q_1)^{n_1}\,, \\
&R_2(n_2) = 1 - (q_2)^{n_2}\,, \\
&c_f = 1000,\ c_1 = 1,\ c_2 = 2\,, \\
&p_1 = 0.75,\ p_2 = 0.8
\end{aligned}$$

subject to:

$$\begin{aligned}
C1: &\quad 1 \le n_1 \le 5 \\
C2: &\quad 1 \le n_2 \le 6 \\
C3: &\quad n_1 + n_2 \le 8 \quad \text{total number constraint} \\
C4: &\quad 2.n_1 + 3.n_2 \le 20 \quad \text{weight constraints} \\
C5: &\quad R_s(n1, n2) \ge 0.9\,.
\end{aligned}$$

Because we know the explicit limits on n_1 and n_2, we can find all possible combinations of the possible solutions. For each possible solution combination, we check the other constraints. If any constraint is violated, we discard that combination and go for the next combination. For all valid combinations, we compute the objective function. The combination that correspond to the minimum value for the objective function is the optimal solution. This is described in the Table 54.1.

The optimal solution is ($n_1 = 4$, $n_2 = 4$) and the corresponding cost is 17.5. If there are no constraints for this problem, the optimal solution is ($n_1 = 5, n_2 = 4$) and the corresponding cost is 15.575.

Although it is easy to understand and program exhaustive searching methods, they are infeasible for large systems. This is because the search space increases exponentially with the problem size. Therefore, researchers proposed methods that can only search within a reduced space. *Misra* [54.20] has proposed an exact algorithm for optimal redundancy allocation problem with a reliability maximization objective based on a search near the boundary of the feasible region. This method was later implemented by *Misra* and *Sharma* [54.28], and *Misra* and *Misra* [54.29] to solve various redundancy allocation problems. This does not always gives an exact optimal solution. *Prasad* and *Kuo* [54.30] recently developed a partial enumeration method based on lexicographic search with an upper bound on system reliability. This method was demonstrated for both small and large problems. An overview of other exact methods can be found in [54.6]. With some minor modifications, the same concepts can also be used for cost-effective design.

Dynamic Programming

Dynamic programming (DP) is a solution methodology based on Richard Bellman's principle of optimality. DP is an approach for solving a wide spectrum of decision-making problems and is highly effective at solving certain types of deterministic and stochastic optimization problems. In this approach, a decision-making problem involving n variables is reduced to a set of n single-variable problems, which are derived sequentially in such a way that the solution of each problem is obtained using the preceding answer. The DP approach transforms an n-variable optimization problem into a multi-stage decision-making process. Such a transformation is not always straightforward and it quite often requires a lot of ingenuity. For this reason, it is not easy to describe the general DP approach as an algorithm. An outline of DP is given for various optimization problems in [54.6].

The basic DP approach is generally used to solve optimization problems with, at most, one constraints and with or without bounds on the decision variables. In this section, we demonstrate the DP approach solving a three-unit series system without constraints and

bounds.

Minimize

$$C_s = n_1c_1 + n_2c_2 + n_3c_3 + c_f[1 - R_s(n_1, n_2, n_3)]$$

where, (54.54)

$$\begin{aligned} R_s(n_1, n_2, n_3) &= R_1(n_1)R_2(n_2)R_3(n_3)\,, \\ R_1(n_1) &= 1 - (q_1)^{n_1}\,, \\ R_2(n_2) &= 1 - (q_2)^{n_2}\,, \\ R_3(n_3) &= 1 - (q_3)^{n_3}\,, \\ c_f &= 10,\ c_1 = 1,\ c_2 = 1,\ c_3 = 0.2\,, \\ p_1 &= 0.75,\ p_2 = 0.5,\ p_3 = 0.33\,. \end{aligned}$$

For example assume that the optimal configurations for stages 2 and 3 are fixed, i. e., n_2 and n_3 are fixed. Therefore, $n_2c_2 + n_3c_3$ and $R_2(n_2)R_3(n_3)$ are also fixed. Let us define v_i as the probability that the stages $i, \cdots, 3$ are working. Hence, the problem is reduced to:

$$\text{Minimize } C_s = n_1c_1 + c_f[1 - v_2R_1(n_1)]\,. \quad (54.55)$$

The solution to this problem depends on the value of v_2. For variable values of v_2, we find optimal value of n_1, n_1^*, using simple search techniques. We can also find the optimal solution analytically. Let the corresponding value for C_s be $f_1(v_2)$. The results are provided in Table 54.2. Now the remaining problem is to find the value of v_2 corresponding to the optimal values of n_2 and n_3. Let us assume that we know the value of n_3. Therefore, we also know $v_3 \equiv R_3(n_3)$. Hence, the problem is reduced to:

Minimize

$$\begin{aligned} C_s &= n_1c_1 + n_2c_2 + c_f[1 - v_3R_1(n_1)R_2(n_2)] \\ &= n_2c_2 + f_1[v_3R_2(n_2)]\,. \end{aligned} \quad (54.56)$$

For any fixed v_3, we can find the value of C_s for varying values of n_2. In this process, we need to compute $f_1(v)$ for given value of x_2. This can be done through interpolation. For example, the value of $f_1(v)$ for $v = 0.88$ is determined by interpolation of $f_1(0.9)$ and $f_1(0.8)$ obtained in the previous stage (see Table 54.2). Once we compute the C_s for various values of n_2, we can find the n_2^* that minimize C_s. Let the corresponding value for C_s for a fixed v_3 be $f_2(v_3)$.

Now the problem is reduced to find the value of v_3 corresponding to the optimal value of n_3. The cost function can be rewritten as

Minimize

$$\begin{aligned} C_s &= n_1c_1 + n_2c_2 + n_3c_3 + c_f[1 - R_s(n_1, n_2, n_3)] \\ &= n_3c_3 + f_2(R_3(n_3)) \end{aligned} \quad (54.57)$$

Table 54.2 Dynamic programming solution

v_2	$n_1^*(v_2)$	$f_1(v_2)$	$n_2^*(v_3)$	$f_2(v_3)$
1.0	2	2.625	1.0	6.997
0.9	2	3.563	0.9	7.617
0.8	2	4.500	0.8	8.375
0.7	2	5.438	0.7	9.031
0.6	2	6.375	0.6	9.625
0.5	1	7.250	0.5	10.125
0.4	1	8.000	0.4	10.500
0.3	1	9.750	0.3	10.875
0.2	1	9.500	0.2	11.250
0.1	1	10.250	0.1	11.125

We can find the value of $R_3(n_3)$ for varying values of n_3. Hence, we can compute $f_2(v)$ and C_s for varying values of n_3. Hence, we can find the value n_3^* that minimizes C_s. The optimal value of n_3 is $n_3^* = 7$, and the corresponding minimum cost is 8.6787. This is the minimum cost for the overall system. From this we can calculate the corresponding $v_3 = R_3(n_3^*) = 0.94$. Using the backtracking method, we can compute $n_2^*(v_3) = n_2^*(0.94) = 3$. Similarly, we can compute $v_2 = 0.82$. The optimal choice for n_1 is $n_1^*(v_2) = n_1^*(0.82) = 2$ and the corresponding v_1 is 0.77.

Therefore, the optimal solution is $(n_1^*, n_2^*, n_3^*) = (2,3,7)$ and the corresponding minimum average cost is 8.6787.

It should be noted that the whole process can be expressed using the following recursive relationship. Let $f_i(v_{i+1})$ denote the minimum expected average profit due to redundancy at stages $1, \cdots, i$, where v_{i+1} is the probability that the stages $i+1, \cdots, 3$ work. Let $v_4 = 1$. Then, for $i = 1, \cdots, 3$, we have the following recursive relationship

$$f_i(v_i + 1) = \min \arg_{n_i}\{f_{i-1}[v_{i+1}R_i(x_i)] + c_ix_i\} \quad (54.58)$$

for $i = 2, 3$. For $i = 3$, it is enough to consider only the value 1.0 for v_{i+1}. Let $n_i{}^*(v_{i+1})$ denote the value of n_i that maximizes $f_{i-1}[v_{i+1}R_i(n_i)] + c_in_i$, then

$$\begin{aligned} f_1(v_2) &= \min \arg_{n_1}\{c_fv_2[1 - (1 - r_1)^{n_1}] + c_1n_1\}, \\ f_2(v_3) &= \min \arg_{n_2}\{f_1(v_3[1 - (1 - r_2)^{n_2}]) + c_2n_2\}, \\ f_3(1.0) &= \min \arg_{n_3}\{f_2[1 - (1 - r_3)^{n_3}] + c_3n_3\}\,. \end{aligned} \quad (54.59)$$

Although recursive relations are available, it is not computationally feasible to evaluate $f_i(v)$ for all v in the

continuous interval [0, 1]. Thus, $f_i(v)$ is evaluated on a grid of v for $i = 1$ and 2. Linear interpolation is done when $f_i(v)$ is needed, but not available, for a required value of v.

54.4 Hybrid Optimization Algorithms

54.4.1 Motivation

The problem of finding the optimal number of spares is known as a redundancy allocation problem. It is a nonlinear programming problem involving integer variables [54.31]. *Chern* [54.18] has shown that the problem is NP-hard even for series–parallel systems. A summary of approaches to determine optimal or near optimal solutions in this area is presented in [54.2, 3]. In most of the published articles, the objective is the maximization of a system performance measure such as reliability or availability that satisfies a set of given constraints. In a cost-effective solution, the objective should be the minimization of the total costs associated with the system, which is the sum of the cost of initial spares, cost of maintenance, and cost of downtime. The publications on cost-effective designs are limited [54.9, 12, 32, 33]. Furthermore, even though the majority of systems are repairable, few publications exist on the optimal design issues of repairable systems [54.12, 21, 33, 34].

In general, it is difficult to obtain the optimal solution for complex systems. There exist several heuristics algorithms [54.35–37] in the literature to find near-optimal solutions. However, the efficiency and computational effort of those methods depend on the lower and upper bounds of the decision variables [54.2]. In this chapter we present a new method to find tighter bounds for obtaining the optimal number of spares. The main contributions of this research is a development of a new method to find tighter bounds for the optimal number of spares for each subsystem. The efficiency of the proposed bounds is demonstrated through several examples.

54.4.2 Rationale for the Hybrid Method

Under some mild assumptions the cost-effective solutions of both repairable and nonrepairable systems can be represented using the same mathematical formulation. For example, the cost-effective formulations used in Sects. 54.1 and 54.2 are in the same form. Therefore, the same optimization methods can used for solving these two problems. Although, the underlying problem is NP-hard, when there is only one subsystem in the system, we can find the exact solution in an efficient way even when spares are in cold- or hot-standby mode. Furthermore, we can find the exact solution even if the subsystem adopt the k-out-of-n configuration. Using our latest research findings, we can show that, when the cost of failure is the same, optimal value obtained for the subsystem in a stand-alone analysis is in fact the upper bound in the entire analysis. The upper bound can be used find the lower bounds. Further, using these lower and upper bounds, we can find better bounds in an iterative way until we reach stable bounds. This new way of finding bounds reduces the search space considerably and hence improves the solution quality.

We first describe this method for repairable systems. Then we apply the same method for the nonrepairable case.

54.4.3 Repairable Systems

Cost Minimization with One Subsystem

Before solving the actual problem, first consider a simpler problem that contains only one subsystem, i. e., $m = 1$. Say it is the subsystem i. Therefore, the cost of the system is

$$T(n_i) = c_i n_i + c_f [1 - A_i(n_i)] \ ,$$
$$A_i(n_i) \equiv \text{binf}(n_i - k_i; q_i, n_i) \ . \qquad (54.60)$$

It should be noted that the exact solution for this problem cannot be found with simple searching methods, where adding spares is stopped once a spares quantity that increases the overall system cost is reached. This is because there can exist multiple discontinuous regions where the average system cost increases with additional spares. For details, refer to [54.32]. Therefore, we need efficient algorithms to solve these optimization problems. Optimal design policies for k-out-of-n systems, i. e., systems with single subsystems, are studied in [54.9, 33].

Definition

$$f(n_i) \equiv A_i(n_i + 1) - A_i(n_i)$$
$$= \binom{n_i}{k_i - 1} p_i^{k_i} q_i^{n_i - k_i + 1} \ ,$$

$$m_0 \equiv \max\left[k_i, \text{gilb}\left(\frac{k_i-1}{p_i}\right)\right] ,$$
$$m_1 \equiv \frac{T(k_i)}{c_i} ,$$
$$m_2 \equiv \inf\{n_i \in [m_0, m_1] : f(n_i) < c_i/c_f\} . \quad (54.61)$$

It should be noted that m_2 can be evaluated efficiently using binary searching methods. When sequential searching is performed, we can use the following relationship to reduce the computational efforts

$$f(n_i) = f(n_i-1)\frac{n_i q_i}{n_i - k_i + 1} ,$$
$$f(k_i - 1) = 1 . \quad (54.62)$$

Theorem 54.1

For fixed k_i, p_i, c_i, and c_f, the optimal value of n_i that minimizes $T(n_i)$ is n_i^*:

if $f(m_0) < c_i/c_f$, then $n_i^* = k_i$
else if $f(k_i) \geq c_i/c_f$, then $n_i^* = m_2$
else if $T(k_i) > T(m_2)$, then $n_i^* = m_2$
else $n_i^* = k_i$.

Proof: The form of the cost function for this problem is equivalent to the form of the cost function used in references [54.9, 33] for minimizing the total cost of a nonrepairable k-out-of-n system. Therefore, the underlying mathematical problem is the same. For details, refer to [54.9, 33]. ■

Corollary 54.1

If the configuration of the subsystem is parallel, i. e., $k_i = 1$, then for fixed p_i, c_i, and c_f, the optimal value of n_i that minimizes $T(n_i)$ is $n_i^* = e_1$

$$e_0 \equiv \frac{\ln\left(\frac{c_i}{c_f \cdot p_i}\right)}{\ln(q_i)} ,$$
$$e_1 \equiv \max[1, \text{gilb}(e_0) + 1] . \quad (54.63)$$

Proof: For parallel systems, $f(n) = q^n p$ [54.33]. Therefore, the rest of the proof is straightforward from Theorem 54.1. It should be noted that Corollary 54.1 is a special case of Theorem 54.1. ■

Corollary 54.2

For fixed k_i, p_i, and c_i, the optimal value of n_i that minimizes $T(n_i)$ is nondecreasing with an increase in c_f.

Proof: Because c_i/c_f is decreasing, $k_i \leq m_2$, and $f(n_i)$ is decreasing in $[m_0, m_1]$, the proof is straightforward. ■

Cost Minimization with Multiple Subsystems

The system consists of several subsystems. Except for one subsystem (say subsystem i), the configurations of these subsystems are fixed. The system availability can be expressed using conditional probabilities based on Shannon's decomposition formula [54.10]. The theorem related to this formula is known as the total probability theorem [54.7]. Let us consider that subsystem i is the pivotal element (also called the key element) in the decomposition; then

$$\begin{aligned} A(\boldsymbol{n}) &= h(A_1, \cdots, A_m) \\ &= A_i h_i^1 + (1 - A_i) h_i^0 \\ &= \left(h_i^1 - h_i^0\right) A_i + h_i^0 . \end{aligned} \quad (54.64)$$

Here h_i^1 and h_i^0 are the conditional availabilities of the system given that subsystem i is operating and nonoperating, respectively. Therefore, the total system cost with n_i components in subsystem i is

$$\begin{aligned} T(n_i) &= \sum_{j=1}^{m} c_j n_j + c_f\left[1 - \left(h_i^1 - h_i^0\right) A_i(n_i) - h_i^0\right] \\ &= c_0 + c_f\left(1 - h_i^1\right) + c_i n_i \\ &\quad + c_f\left(h_i^1 - h_i^0\right)\left[1 - A_i(n_i)\right] , \\ c_0 &= \sum_{j=1;\, j\neq i}^{m} c_j n_j . \end{aligned} \quad (54.65)$$

Because $\left[c_0 + c_f\left(1 - h_i^1\right)\right]$ is independent of n_i, minimizing $T(n_i)$ in (54.65) is equivalent to minimizing $T_1(n_i)$ in (54.66)

$$T_1(n_i) = c_i n_i + c_f\left(h_i^1 - h_i^0\right)\left[1 - A_i(n_i)\right] . \quad (54.66)$$

The cost equations (54.60) and (54.66) are similar. Therefore, we can find the optimal n_i using Theorem 54.1. However, instead of using c_f, we need to use the modified system cost $c_f\left(h_i^1 - h_i^0\right)$ in Theorem 54.1. The results are presented in Corollary 54.3.

Definition

$$m_2' \equiv \inf\left\{n_i \in [m_0, m_1] : f(n_i) < \frac{c_i}{c_f\left(h_i^1 - h_i^0\right)}\right\} . \quad (54.67)$$

Corollary 54.3

For fixed configurations of all subsystems (except subsystem i) and for fixed k_i, p_i, c_i, and c_f, the optimal value of n_i that minimizes $T(n_i)$ is n_i^*:

$$\begin{aligned}&\text{if } f(m_0) < \frac{c_i}{c_f(h_i^1 - h_i^0)}, \text{ then } n_i^* = k_i\\ &\text{else if } f(k_i) \geq \frac{c_i}{c_f(h_i^1 - h_i^0)}, \text{ then } n^* = m_2'\\ &\qquad \text{else if } T(k_i) > T(m_2), \text{ then } n_i^* = m_2'\\ &\qquad\qquad \text{else } n_i^* = k_i .\end{aligned}$$

Proof: The proof is straightforward from Theorem 54.1. ■

Corollary 54.4

If all subsystems are in series and the configuration of each subsystem is parallel, then for fixed configurations of all subsystems (except subsystem i) and for fixed h_i^1, p_i, c_i, and c_f, the optimal value of n_i that minimizes $T(n_i)$ is $n_i^* = e_3$:

$$e_2 \equiv \frac{\ln\left(\frac{c_i}{c_f h_i^1 p}\right)}{\ln(q)} ,$$

$$e_3 \equiv \max[1, \text{gilb}(e_2) + 1] . \tag{54.68}$$

Proof: For series systems: $h_i^0 = 0$ and $A_i(n_i) = 1 - q_i^{n_i}$. Hence, the rest of the proof follows from Corollaries 1 and 3. ■

It should be noted that Corollary 54.4 is also applicable even if all subsystems are not in series. The only requirement is that subsystem i is in series with the rest of the system.

Corollary 54.5

The optimal n that minimizes the average system cost is always less than or equal to the n that minimizes average cost of the subsystem with the same failure cost.

Proof: Because $c_f(h_i^1 - h_i^0) \leq c_f$, from Corollary 54.2, the proof is straightforward. ■

Simultaneous Optimization

It is well known that simultaneous optimization of the components in all subsystems is a nonlinear integer programming problem consisting of a vector of decision variables. It is important to know the bounds on the decision variables for applying general-purpose optimization algorithms such as GA or SA. The efficiency of optimization algorithms, in terms of computational time as well as solution quality, can be improved by reducing the search space [54.6].

It is reasonable to assume that all subsystems must contain at least the minimum number of components that are required for its successful operation. This assumption is valid as long as we are not allowed to change the system configuration by removing some subsystems. In this chapter, we assume that we are not allowed to change the system configuration and subsystem i must have at least k_i components. Therefore, $n_i^L = k_i$; i. e., the default lower bound is $\boldsymbol{n}^L = \boldsymbol{n}^K$.

Furthermore, it is well known that the maintenance cost of all components should be less than or equal to the cost of downtime. Therefore,

$$\sum_{i=1}^{m} n_i c_i \leq c_f . \tag{54.69}$$

In the worst case, the above equation can lead to the following upper bound on n_i^* [54.33]

$$n_i^U = \text{gilb}\left(\frac{c_f}{c_i}\right) . \tag{54.70}$$

However, a better upper bound [54.33] could be:

$$n_i^U = k_i + \text{gilb}\left(\frac{c_f[1 - A(\boldsymbol{n}^K)]}{c_i}\right) . \tag{54.71}$$

Within a given set of lower and upper bounds, the total number of solution vectors in the search space is equivalent to M

$$M = \prod_{i=1}^{m} \left(n_i^U - n_i^L + 1\right) . \tag{54.72}$$

In general, these lower and upper bounds are very wide (refer to the examples). At the same time, there seems to be no method that finds the better bounds in a systematic way. In this chapter, we show that the results presented in the previous sections can be used to find tighter bounds that reduce the search space and hence increase the efficiency of the solution procedure.

Upper Bound. According to Corollary 54.5, the upper bound on n_i^* (i. e., n_i^U) that minimizes the average system cost is always less than or equal to the n_i^* that minimizes the average cost of the subsystem with the same failure cost. Therefore, the optimal value obtained using Theorem 54.1 is an upper bound for n_i^*. An improved upper bound can be found using Theorem 54.2.

Theorem 54.2

If $\boldsymbol{n}^L$ is a lower bound for $\boldsymbol{n}^*$ and $\boldsymbol{n}^U$ is an upper bound for $\boldsymbol{n}^*$, then the improved upper bound on n_i^* is equal

to $n_i^{U'}$.

$$
\begin{aligned}
n_i^{U'} &\equiv \text{optimal value obtained using Corollary 54.3 with the following parameters,}\\
h_i^1 &\equiv h_i^1(\boldsymbol{n}^{\mathrm{U}}) = \text{conditional availability with } \boldsymbol{n}^{\mathrm{U}} \text{ given that subsystem } i \text{ is operating}\\
&\equiv h\left(A_1^{\mathrm{U}}, \cdots, A_i^{\mathrm{U}} = 1, \cdots, A_m^{\mathrm{U}}\right) ,\\
A_j^{\mathrm{U}} &\equiv \mathrm{binf}(n_j - k_j; q_j, n_j^{\mathrm{U}}) ,\\
h_i^0 &\equiv h_i^0(\boldsymbol{n}^{\mathrm{L}}) = \text{conditional availability with } \boldsymbol{n}^{\mathrm{L}} \text{ given that subsystem } i \text{ is nonoperating}\\
&\equiv h\left(A_1^{\mathrm{L}}, \cdots, A_i^{\mathrm{L}} = 0, \cdots, A_m^{\mathrm{L}}\right) ,\\
A_j^{\mathrm{L}} &\equiv \mathrm{binf}\left(n_j - k_j; q_j, n_j^{\mathrm{L}}\right) .
\end{aligned}
\tag{54.73}
$$

Proof: Because the system is coherent, $h_i^1(\boldsymbol{n}^{\mathrm{U}})$ is greater than or equal to $h_i^1(\boldsymbol{n}^*)$, and $h_i^0(\boldsymbol{n}^{\mathrm{L}})$ is less than or equal to $h_i^0(\boldsymbol{n}^*)$. Therefore, $(h_i^1 - h_i^0)$ at the optimal point is always less than or equal to $[h_i^1(\boldsymbol{n}^{\mathrm{U}}) - h_i^0(\boldsymbol{n}^{\mathrm{L}})]$. From Corollary 54.2, the optimal value is decreasing with a decrease in c_{f} (the effective c_{f}). Therefore, n_i^{U}, obtained from Theorem 54.2 using Corollary 54.3, is the better upper bound. ■

Lower Bound. The default lower bound is $\boldsymbol{n}^{\mathrm{L}} = \boldsymbol{n}^{\boldsymbol{K}}$. The upper bound $\boldsymbol{n}^{\mathrm{U}}$ can be computed using the procedure presented in Sect. 54.4.3. In this section, we provide an improved lower bound for a given set of previous lower and upper bounds: $\boldsymbol{n}^{\mathrm{L}}$ and $\boldsymbol{n}^{\mathrm{U}}$.

Theorem 54.3

If $\boldsymbol{n}^{\mathrm{L}}$ is a lower bound for $\boldsymbol{n}^*$ and $\boldsymbol{n}^{\mathrm{U}}$ is an upper bound for $\boldsymbol{n}^*$, then the improved lower bound on n_i^* is equal to $n_i^{L'}$

$$
\begin{aligned}
n_i^{L'} &\equiv \inf\{n_i \geq n_i^{\mathrm{L}} : A_i(n_i) \geq B_i\}\\
B_i &\equiv \frac{A(\boldsymbol{n}^{\mathrm{U}}) - h_i^0(\boldsymbol{n}^{\mathrm{U}}) - \frac{(C_{\mathrm{U}} - C_{\mathrm{L}})}{c_{\mathrm{f}}}}{h_i^1(\boldsymbol{n}^{\mathrm{U}}) - h_i^0(\boldsymbol{n}^{\mathrm{L}})} .
\end{aligned}
\tag{54.74}
$$

Proof: It should be noted that the average cost of the system at the optimal point should always be less than or equal to the cost of the system at any other $\boldsymbol{n}$. Hence,

$$
T(\boldsymbol{n}^*) \leq T(\boldsymbol{n}^{\mathrm{U}}) = C_{\mathrm{U}} + c_{\mathrm{f}} U(\boldsymbol{n}^{\mathrm{U}}) . \tag{54.75}
$$

Maintenance cost is a strictly increasing function in $\boldsymbol{n}^*$. Therefore, $C_{\mathrm{L}} \leq T_{\mathrm{m}}(\boldsymbol{n}^*)$. Hence,

$$
\begin{aligned}
&C_{\mathrm{L}} + c_{\mathrm{f}} U(\boldsymbol{n}^*) \leq C_{\mathrm{U}} + c_{\mathrm{f}} U(\boldsymbol{n}^{\mathrm{U}}) ,\\
&\left[h_i^1(\boldsymbol{n}^*) - h_i^0(\boldsymbol{n}^*)\right] A_i(n_i^*) + h_i^0(\boldsymbol{n}^*)\\
&\qquad \geq A(\boldsymbol{n}^{\mathrm{U}}) - \frac{(C_{\mathrm{U}} - C_{\mathrm{L}})}{c_{\mathrm{f}}} .
\end{aligned}
\tag{54.76}
$$

Because the system is coherent, we have

$$
\begin{aligned}
&h_i^0(\boldsymbol{n}^{\mathrm{U}}) \geq h_i^0(\boldsymbol{n}^*) ,\\
&h_i^1(\boldsymbol{n}^{\mathrm{U}}) - h_i^0(\boldsymbol{n}^{\mathrm{L}}) \geq h_i^1(\boldsymbol{n}^*) - h_i^0(\boldsymbol{n}^*) .
\end{aligned}
\tag{54.77}
$$

Hence,

$$
A_i(n_i^*) \geq \frac{A(\boldsymbol{n}^{\mathrm{U}}) - h_i^0(\boldsymbol{n}^{\mathrm{U}}) - \frac{(C_{\mathrm{U}} - C_{\mathrm{L}})}{c_{\mathrm{f}}}}{h_i^1(\boldsymbol{n}^{\mathrm{U}}) - h_i^0(\boldsymbol{n}^{\mathrm{L}})} \equiv B_i . \tag{54.78}
$$

Because $A_i(n_i^*)$ is an increasing function n_i^*, the new lower bound is

$$
n_i^{L'} \equiv \inf[n_i : A_i(n_i) \geq B_i] . \tag{54.79}
$$

■

Corollary 54.6

If $\boldsymbol{n}^{\mathrm{L}}$ is a lower bound for $\boldsymbol{n}^*$, $\boldsymbol{n}^{\mathrm{U}}$ is an upper bound for $\boldsymbol{n}^*$, and the configuration of subsystem i is parallel, then the improved lower bound on n_i^* is equal to $n_i^{L'}$.

$$
\begin{aligned}
n_i^{L'} &\equiv \max[1, \mathrm{gilb}(L_i) + 1] ,\\
L_i &\equiv \frac{\ln(1 - B_i)}{\ln(q_i)} .
\end{aligned}
\tag{54.80}
$$

B_i is defined in (54.74).

Proof: For parallel subsystems, $A_i(n_i^*) = 1 - q_i^{n_i^*}$. Hence, the rest of the proof is straightforward from Theorem 54.3.

If the subsystem i is in series with the rest of the system, then the equation for B_i can be simplified as shown in (54.81).

$$
B_i \equiv A_i\left(n_i^{\mathrm{U}}\right)\left(1 - \frac{(C_{\mathrm{U}} - C_{\mathrm{L}})}{c_{\mathrm{f}} A(\boldsymbol{n}^{\mathrm{U}})}\right) . \tag{54.81}
$$

■

Algorithm

1. Find the default lower bound. It is $\boldsymbol{n}^{\mathrm{L}} = \boldsymbol{n}^{\boldsymbol{K}} = [k_1, \cdots, k_m]$.
2. Find the optimal value for each subsystem using Theorem 54.1, considering that there is only one subsystem in the system and the cost of failure is c_{f}. Let the optimal value for subsystem i be n_i^{U}. From Corollary 54.5, it is the upper bound for n_i^*.
3. Find $A_i(n_i^{\mathrm{L}})$ and $A_i(n_i^{\mathrm{U}})$ for each subsystem.
4. Find $A(\boldsymbol{n}^{\mathrm{L}})$ and $A(\boldsymbol{n}^{\mathrm{U}})$.
5. Find $h_i^0(n_i^{\mathrm{L}})$, $h_i^0(n_i^{\mathrm{U}})$, $h_i^1(n_i^{\mathrm{L}})$, and $h_i^1(n_i^{\mathrm{U}})$ for each subsystem. For series configurations, we

have: $h_i^0(n_i^{\mathrm{L}}) = 0$, $h_i^0(n_i^{\mathrm{U}}) = 0$, $h_i^1(n_i^{\mathrm{L}}) = \frac{A(\boldsymbol{n}^{\mathrm{L}})}{A_i(n_i^{\mathrm{L}})}$, and $h_i^1(n_i^{\mathrm{U}}) = \frac{A(\boldsymbol{n}^{\mathrm{U}})}{A_i(n_i^{\mathrm{U}})}$.

6. Using Theorem 54.2, find the updated $\boldsymbol{n}^{\mathrm{U}}$.
7. Using Theorem 54.3, find the updated $\boldsymbol{n}^{\mathrm{L}}$.
8. If stable lower and upper bounds are reached, i. e., if there are no change in the lower and upper bounds, then continue to Step 9. Otherwise (if there is a change), repeat from Step 3.
9. Using n_i^{L} and n_i^{U} as the bounds for optimal n_i, find the optimal vector that minimizes $T(\boldsymbol{n})$.
   - We can use any search algorithm for optimization including GA [54.38, 39], SA [54.40], or any other general-purpose optimization algorithms [54.35–37, 41].
   - In this chapter, we use an exhaustive searching algorithm to illustrate the example problems.

Numerical Examples

Series System with Parallel Subsystems. Consider a system consisting of five parallel subsystems arranged in a series configuration. The reliability block diagram of the system is shown in Fig. 54.1.

The failure-time distribution of a component in subsystem i is Weibull with characteristic life η_i and shape parameter β_i. There exist several forms of probability density function (PDF) for the Weibull distribution. The form of the Weibull distribution considered in this chapter is

$$f_i(t) = \frac{\beta_i}{\eta_i}\left(\frac{t}{\eta_i}\right)^{\beta_i - 1} \exp\left[-\left(\frac{t}{\eta_i}\right)^{\beta_i}\right] . \qquad (54.82)$$

Hence, the MTTF of each component (ϕ_i) is

$$\phi_i = \eta_i \Gamma(1/\beta_i + 1) . \qquad (54.83)$$

The repair-time distribution of a component in subsystem i is lognormal with parameters μ_i and σ_i. Hence,

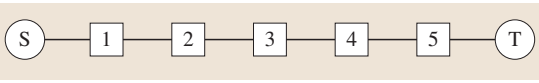


Fig. 54.1 Reliability block diagram for a series system

Table 54.4 Parameters for optimization of a series system

i	p_i	q_i	ν_i	R_i	c_i
1	0.90886	0.09114	0.00091	109.98298	0.10026
2	0.86842	0.13158	0.00174	110.67862	0.19272
3	0.97501	0.02499	0.00049	131.81396	0.06471
4	0.96746	0.03254	0.00129	83.27819	0.10706
5	0.79644	0.20356	0.00159	393.22117	0.62662

the MTTR of each component (ψ_i) is

$$\psi_i = \exp\left(\mu_i + \frac{\sigma_i^2}{2}\right) . \qquad (54.84)$$

In this example, the MLDT associated with repairs is considered to be zero for all components. Using the system parameters in Table 54.3, we can find the other parameters of the components in each subsystem. These parameters include availabilities (p_i), unavailabilities (q_i), failure frequencies (ν_i), cost of each repair (R_i), and repair cost per unit time (c_i). Values for these parameters are provided in Table 54.4.

1. Because the configuration of the system is series, the system availability expression can be obtained using (54.1).
2. Without the results of this chapter, $\boldsymbol{n}^{\mathrm{L}} = \{1, 1, 1, 1, 1\}$ and $\boldsymbol{n}^{\mathrm{U}} = \{4060, 2113, 6291, 3802, 650\}$. Therefore, the search space contains more than 1.3×10^{17} solution vectors. It is unrealistic to search for an exact solution within such a large solution space.
3. Using common-sense reasoning, we may guess that the actual optimal solution will be somewhere between the following bounds: $\boldsymbol{n}^{\mathrm{L}} = \{1, 1, 1, 1, 1\}$ and

Table 54.3 Parameters for a series system

Subsystem ID	Failure distribution Weibull			Repair distribution Lognormal			Repair cost Parameters	
i	η_i	β_i	ϕ_i	μ_i	σ_i	ψ_i	γ_i	δ_i
1	1125	2	997.01	4.00	1.10	99.98	10.00	1.00
2	540	1.3	498.73	3.20	1.50	75.57	20.00	1.20
3	2200	1.5	1986.04	3.75	0.60	50.91	30.00	2.00
4	800	1.2	752.52	3.20	0.25	25.31	20.00	2.50
5	475	0.9	499.79	4.35	1.00	127.74	10.00	3.00

Cost per unit downtime (c_{f}) = 1000 cost units

$\boldsymbol{n}^{\mathrm{U}} = \{100, 100, 100, 100, 100\}$. Even this search space contains 10^{10} solution vectors. To reduce the search space, we may guess that the upper bound might be $\boldsymbol{n}^{\mathrm{U}} = \{10, 10, 10, 10, 10\}$. This contains 10^5 solution vectors. To reduce the search space even further, we might guess that $\boldsymbol{n}^{\mathrm{U}} = \{4, 4, 4, 4, 4\}$. This search space contains 1024 solution vectors. Now, if we can compute the objective function (cost function) for each vector, we can find the exact solution. However, if the optimal vector is not within the assumed lower and upper bounds, then we will not obtain correct results. Therefore, a systematic procedure to find the bounds is essential.

4. Using Theorem 54.1 (Corollary 54.1) and Corollary 54.5, $\boldsymbol{n}^{\mathrm{U}} = \{4, 5, 3, 3, 5\}$. Even without applying the proposed lower bound, i.e, even with $\boldsymbol{n}^{\mathrm{L}} = \{1, 1, 1, 1, 1\}$, we have 900 solution vectors. The number of solution vectors using this upper bound is less than the number of solution vectors of the above wild guess (which leads to incorrect results).
5. Further, using Theorem 54.2 (Corollary 54.6), we have $\boldsymbol{n}^{\mathrm{L}} = \{3, 3, 2, 2, 4\}$. Therefore, now there are only 48 solution vectors within the bounds.
6. With exhaustive searching, the optimal vector of components that minimizes $T(\boldsymbol{n})$ is $\boldsymbol{n}^* = \{4, 5, 3, 3, 5\}$. For this example, $\boldsymbol{n}^* = \boldsymbol{n}^{\mathrm{U}}$. The corresponding system availability is 0.99949, and the corresponding minimum cost per unit time is 5.521.

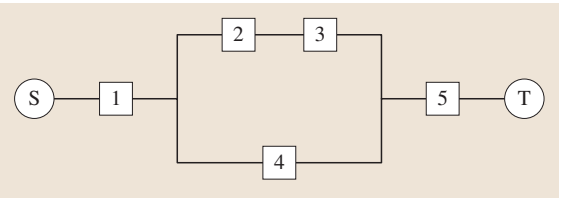


Fig. 54.2 A hypothetical reliability block diagram

Hypothetical Reliability Block Diagram. Consider a hypothetical example where subsystems are connected using the reliability block diagram shown in Fig. 54.2. The failure and repair times of each component follow exponential distributions. The logistic delay follows a deterministic time distribution. The parameters of the system are shown in Table 54.5. The parameters required for optimization are provided in Table 54.6.

1. The expression for the availability as a function of the subsystems availabilities can be obtained as follows.

$$\begin{aligned} A(\boldsymbol{n}) &= h(A_1, A_2, A_3, A_4, A_5) \\ &= A_1 A_5 \left[1 - (1 - A_2 A_3)(1 - A_4)\right] , \end{aligned} \tag{54.85}$$

   where A_i can be calculated using (54.51).
2. The default lower bound for n_i^* is $n_i^{\mathrm{L}} = k_i$. Therefore, $\boldsymbol{n}^{\mathrm{L}} = \{2, 3, 2, 3, 2\}$.
3. Without using the results of this chapter, $\boldsymbol{n}^{\mathrm{U}} = \{2860, 1472, 1889, 7765, 1386\}$. This pro-

Table 54.5 Parameters for a hypothetical reliability block diagram

Subsystem ID	Required #	MTTF/MTTR/MLDT			Repair cost	
i	k_i	ϕ_i	ψ_i	φ_i	γ_i	δ_i
1	2	1000	90	10	10.00	1.20
2	3	500	50	25	20.00	2.00
3	2	2000	40	10	30.00	2.50
4	3	750	20	5	20.00	3.00
5	2	500	100	25	10.00	3.50
Cost per unit downtime (c_{f}) = 1000 cost units						

Table 54.6 Parameters for the optimization of a hypothetical reliability block diagram

i	k_i	p_i	q_i	v_i	R_i	c_i
1	2	0.90886	0.09114	0.00091	109.98298	0.10026
2	3	0.86842	0.13158	0.00174	110.67862	0.19272
3	2	0.97501	0.02499	0.00049	131.81396	0.06471
4	3	0.96746	0.03254	0.00129	83.27819	0.10706
5	2	0.79644	0.20356	0.00159	393.22117	0.62662

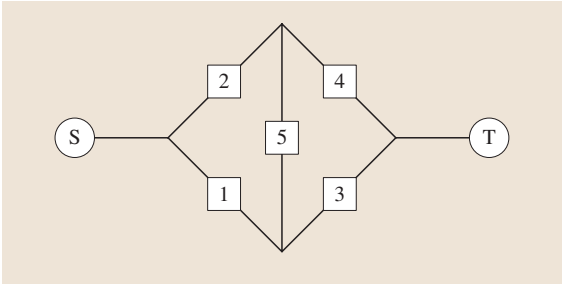


Fig. 54.3 A repairable bridge network

duces more than 8.5×10^{16} solution vectors (points of the search space).
4. Using Theorem 54.1 and Corollary 54.5, $\boldsymbol{n}^{\mathrm{U}} = \{6, 8, 5, 5, 5\}$. At present, the search space contains 1440 solution vectors.
5. Within two iterations, we obtain the stable $\boldsymbol{n}^{\mathrm{L}} = \{5, 3, 2, 3, 4\}$ and $\boldsymbol{n}^{\mathrm{U}} = \{6, 6, 4, 5, 5\}$. Now the search space contains 144 solution vectors.
6. With exhaustive searching, the optimal vector of components that minimizes $T(\boldsymbol{n})$ is $\boldsymbol{n}^* = \{6, 3, 2, 5, 5\}$. The corresponding system availability is 0.9998, and the corresponding minimum cost is 3.1065.

We observe that, in most cases, if a subsystem is in series with the rest of the system, then its optimal value is equivalent or very close to its upper bound. Therefore, using this information, we can further reduce the search space. Hence, the effective search space contains only 36 solution vectors.

Bridge Network. Consider a hypothetical example where subsystems are connected as a bridge network as shown in Fig. 54.3. The parameters used in Table 54.5 and Table 54.6 are considered for this example.

1. The expression for the availability as a function of the subsystem availabilities can be obtained using (54.86).

$$\begin{aligned} A(\boldsymbol{n}) &= h(A_1, A_2, A_3, A_4, A_5) \\ &= A_1A_3 + A_2A_4(1 - A_1A_3) \\ &\quad + A_1A_4A_5U_2U_3 + A_2A_3A_5U_1U_4 \,, \end{aligned} \tag{54.86}$$

where A_i can be calculated using (54.51).
2. The default lower bound for n_i^* is $n_i^{\mathrm{L}} = k_i$. Therefore, $\boldsymbol{n}^{\mathrm{L}} = \{2, 3, 2, 3, 2\}$.
3. Using Theorem 54.1 and Corollary 54.5, $\boldsymbol{n}^{\mathrm{U}} = \{6, 8, 5, 5, 5\}$. At present, the search space contains 1440 solution vectors.
4. Within one iteration, we obtain the stable $\boldsymbol{n}^{\mathrm{L}} = \{2, 3, 2, 3, 2\}$ and $\boldsymbol{n}^{\mathrm{U}} = \{5, 7, 4, 5, 4\}$. Now the search space contains 540 solution vectors.
5. With exhaustive searching, the optimal vector of components that minimizes $T(\boldsymbol{n})$ is $\boldsymbol{n}^* = \{5, 3, 4, 3, 2\}$. The corresponding system availability is 0.9998, and the corresponding minimum cost is 2.5631.

54.4.4 Nonrepairable Systems

The same algorithms that are used for repairable systems can also be used for nonrepairable systems. However, the individual subsystems should be solved using different methods.

The general form for the cost of the single subsystem is

$$\begin{aligned} &T(n_i) = c_i n_i + c_{\mathrm{f}}\left[1 - R(n_i)\right] \,, \\ &R(n_i) \equiv R_i(n_i) \,. \end{aligned} \tag{54.87}$$

Optimal design policies for k-out-of-n systems, i. e., systems with single subsystems, are studied in [54.9,25,33]. Theorem 54.1 presents optimal design policies for k-out-of-n cold/hot-standby systems. Let

Case 1: For hot-standby systems:

$$\begin{aligned} f(n_i) &\equiv \binom{n_i}{k_i - 1} p_i^{k_i} q_i^{n_i - k_i + 1} \\ m_0 &\equiv \max\left[k_i, \mathrm{gilb}\left(\frac{k_i - 1}{p_i}\right)\right] \,; \end{aligned} \tag{54.88}$$

Case 2: For cold-standby systems:

$$\begin{aligned} f(n_i) &\equiv \exp(-k_i\lambda_i t) k_i \lambda_i \frac{(k_i\lambda_i t)^{n_i - k_i}}{(n_i - k_i)!} \,, \\ m_0 &\equiv \max\left[k_i, \mathrm{gilb}\left(k_i\lambda_i t + k_i - 1\right)\right] \,; \end{aligned} \tag{54.89}$$

Definition

$$\begin{aligned} m_1 &\equiv \frac{T(k_i)}{c_i} \,, \\ m_2 &\equiv \inf\{n_i \in [m_0, m_1] : f(n_i) < c_i/c_{\mathrm{f}}\} \,. \end{aligned} \tag{54.90}$$

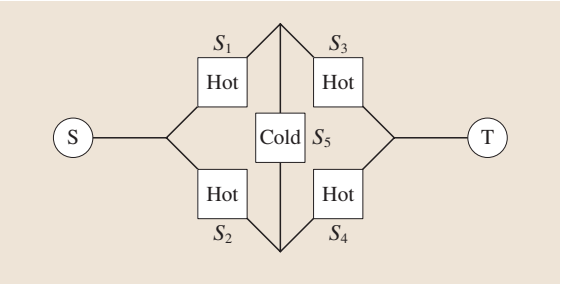


Fig. 54.4 A non-repairablebridge network

Theorem 54.4
For fixed k_i, p_i, c_i, and c_f, the optimal value of n_i that minimizes $T(n_i)$ is n_i^*:
if $f(m_0) < c_i/c_f$, then $n_i^* = k_i$
else if $f(k_i) \geq c_i/c_f$, then $n_i^* = m_2$
else if $T(k_i) > T(m_2)$, then $n_i^* = m_2$
else $n_i^* = k_i$.

Proof: Refer to [54.9, 33]. ■

Numerical Examples

Bridge Network. Consider a hypothetical example where subsystems are connected as a bridge network (Fig. 54.4). The parameters of the system are shown in Table 54.7.

1. The lower bound for n_i^* is $n_i^L = k_i$. Therefore, $\boldsymbol{n}^L = \{2, 3, 3, 2, 3\}$.
2. Without using the results of this chapter, i. e., from reference [54.32], $\boldsymbol{n}^U$ is $\{665, 334, 268, 223, 192\}$. This produces more than 2.4×10^{12} solution vectors. Therefore, exhaustive searching is almost impossible.
3. Using Theorem 54.1 and Corollary 54.3, $\boldsymbol{n}^U$ is $\{6, 6, 7, 5, 8\}$. At present, the search space contains 2, 400 solution vectors. The search space can be reduced further by using the proposed iterative procedure.
4. Within two iterations, we obtain the stable $\boldsymbol{n}^U = \{5, 6, 6, 4, 6\}$. Now the search space contains 768 solution vectors.
5. With exhaustive searching, the optimal vector of components that minimizes $T(\boldsymbol{n})$ is $\boldsymbol{n}^* = \{5, 3, 6, 2, 3\}$. The corresponding system reliability is 0.9998, and the corresponding minimum cost is 44.7640.

Table 54.7 Parameters for a bridge network

Subsystem	k_i	Standby	λ_i	p_i	c_i
S_1	2	hot	0.0001	0.904837	1
S_2	3	hot	0.00005	0.951229	2
S_3	3	hot	0.0001	0.904837	2.5
S_4	2	hot	0.00005	0.951229	3
S_5	3	cold	0.0002	0.818731	3.5

Mission time (t) = 1000 time units
Cost per unit downtime (c_f) = 10 000 cost units

The algorithm presented in this chapter reduced the search space and allowed us to use exhaustive searching to find the optimal component vector.

Series System. Consider a hypothetical space vehicle consisting of the following subsystems arranged in a series configuration:

- guided control subsystem consisting of computers,
- monitoring subsystem consisting of sensors,
- power-generating subsystem.

1. The lower bound for n_i^* is $n_i^L = k_i$. Therefore, $\boldsymbol{n}^L$ is $\{2, 3, 2\}$.
2. Because it is a series system, $h_i^0 = 0$ for all i.
3. Using Theorem 54.1 and Corollary 54.3, $\boldsymbol{n}^U$ is $\{6, 13, 10\}$.
4. In this example, with exhaustive searching, the optimal vector of components that minimizes $T(\boldsymbol{n})$ is $\boldsymbol{n}^* = \{6, 13, 10\}$, which is equivalent to $\boldsymbol{n}^U$. The corresponding system reliability is 0.9834, and the corresponding minimum cost is 139.63.

54.4.5 Conclusions

The proposed lower and upper bounds are easy to find. The computational complexity to find these bounds is linearly proportional to the number of subsystems (m). The lower and upper bounds reduce the search space drastically. Therefore, within a practical time frame, high-quality optimal solutions can be found.

The numerical results demonstrate that, if a subsystem is in series with the rest of the system, then the optimal value for that subsystem is generally equivalent to its upper bound. Therefore, the configuration of such a subsystem can be considered as fixed while performing the simultaneous optimization. This reduces the search space further. For example, in a series system, all subsystems are in series with the rest of the system. Therefore, the optimal solution vector matches with the vector of the upper bounds.

Computational Advantages

One of the main contributions of this chapter is providing the lower and upper bounds on the optimal number of spares in each subsystem. The upper bound for parallel subsystems can be found in constant time using Corollary 54.1. Therefore, the computational complexity of finding the upper bound for this case is O(1). For the general case, the upper bound can be found from Theorem 54.1 using binary searching methods. Therefore, the computational complexity for the general case in finding the upper bound is O[log(m_1)], which is equivalent to O[log(k_i)].

1. For series systems with parallel subsystems, it is known that the near-optimal solution, which is the optimal solution in most cases, is equivalent to its upper bound. Because there are m subsystems, the computation complexity is O(m).
2. For series systems with k-out-of-n subsystems, it is known that the near-optimal solution, which is the optimal solution in most cases, is equivalent to its upper bound. Because there are m subsystems, the computation complexity is O[m log(k)].
3. For the general case, where k-out-of-n type subsystems are arranged in a complex network configuration, the tighter bounds on the decision variable reduces the search space exponentially. Let $r < 1$ be the reduction factor in the width of the bounds obtained by using the proposed tighter bounds with respect to some existing wider bounds on the decision variables [54.33]. Then, the search space is reduces by r^m times. If stochastic algorithms are used to find the optimal solution, then the chances of finding the exact solution will be high with the reduced search space. Let S_o be the search space with some existing bounds, and S_r be the search space with the proposed bounds. Here $S_r = r^m S_o$. Now, randomly select M vectors from the search space. With the original search space, we can find the exact solution with probability $\frac{M}{S_o}$. With the reduced search space, we can find the exact solution with probability $\frac{M}{S_r} \equiv \frac{M}{r^m S_o}$. Therefore, the chances of finding the exact solution increases exponentially. However, evolutionary algorithms, such as GA and SA, find the exact solutions with high probability compared to pure random algorithms. Therefore, the advantage may not be exponential in this case. However, the reduced search space always increase the chances of finding better-quality solutions. If the iterations are not fixed, the same quality of solutions can be obtained quickly with the reduced search space. Our further research will be in the experimental comparison of various algorithms with and without using the proposed bounds.

References

54.1 H. A. Taha: *Operations Research: An Introduction*, 6th edn. (Prentice-Hall of India, New Delhi 2000)

54.2 W. Kuo, V. R. Prasad: An annotated overview of system-reliability optimization, IEEE Trans. Reliab. **49**, 176–187 (2000)

54.3 K. B. Misra: On optimal reliability design: A review, Int. J. Syst. Sci. **12**, 5–30 (1986)

54.4 F. A. Tillman, C. L. Hwang, W. Kuo: Optimization techniques for system reliability with redundancy: a review, IEEE Trans. Reliab. **R26**, 148–155 (1977)

54.5 S. M. Ross: *Introduction to Probability Models*, 8th edn. (Academic, New York 2003)

54.6 W. Kuo, V. R. Prasad, F. A. Tillman, C. Hwang: *Optimal Reliability Design* (Cambridge Univ. Press, Cambridge 2001)

54.7 K. B. Misra: *Reliability Analysis and Prediction: A Methodology Oriented Treatment* (Elsevier, Amsterdam 1992)

54.8 S. V. Amari, H. Pham: *Optimal cost-effective design of complex systems*, Tenth ISSAT International Conference on Reliability and Quality in Design (ISSAT (ISSAT, New Brunswick, NJ 2004) pp. 320–324

54.9 H. Pham: On the optimal design of k-out-of-n: G subsystems, IEEE Trans. Reliab. **41**, 572–574 (1992)

54.10 Y. Chang, S. V. Amari, S. Kuo: Computing system failure frequencies and reliability importance measures using OBDD, IEEE Trans. Comp. **53**, 54–68 (2003)

54.11 S. V. Amari, J. B. Dugan, R. B. Misra: Optimal reliability of systems subject to imperfect fault-coverage, IEEE Trans. Reliab. **48**, 275–284 (1999)

54.12 D. S. Bai, W. Y. Yun, S. W. Cheng: Redundancy optimization of k-out-of-n:G systems with common-cause failures, IEEE Trans. Reliab. **40**, 56–59 (1991)

54.13 L. Nordmann, H. Pham: Reliability of decision making in human-organizations, IEEE Trans. Syst. Man Cyber. A **27**, 543–549 (1997)

54.14 H. Pham: Optimal cost-effective design of triple-modular-redundancy-with-spares systems, IEEE Trans. Reliab. **42**, 369–374 (1993)

54.15 H. Pham, W. J. Galyean: Reliability analysis of nuclear fail-safe redundancy, Reliab. Eng. Syst. Safety **37**, 109–112 (1992)

54.16 H. Pham: Optimal System-Profit Design of Series-Parallel Systems With Multiple Failure Modes, Reliab. Eng. Syst. Safety J. **37**, 151–155 (1992)

54.17 H. Pham: Optimal Design of Parallel-Series Systems With Competing Failure Modes, IEEE Trans. Reliab. **41**, 583–587 (1992)

54.18 M. S. Chern: On the computational complexity of reliability redundancy allocation in a series system, Oper. Res. Let. **11**, 309–315 (1992)

54.19 D. Coit, A. E. Smith: Reliability optimization of series–parallel systems using a genetic algorithm, IEEE Trans. Reliab. **45**, 254–260 (1996)

54.20 K. B. Misra: A simple approach for constrained redundancy optimization problems., IEEE Trans. Reliab. **R21**, 30–34 (1972)

54.21 V. K. Srivastava, A. Fahim: *k*-out-of-*m* system availability with minimum-cost allocation of spares, IEEE Trans. Reliab. **37**, 287–292 (1988)

54.22 Z. W. Birnbaum, J. D. Esary, S. C. Saunders: Multi-component systems and structures and their reliability, Technometrics **3**, 55–77 (1961)

54.23 S. V. Amari, J. B. Dugan, R. B. Misra: A separable method for incorporating imperfect fault-coverage into combinatorial models, IEEE Trans. Reliab. **48**, 267–274 (1999)

54.24 J. Rupe, W. Kuo: Performability of systems based on renewal process models, IEEE Trans. Syst. Man Cyber. A **28**, 691–698 (1998)

54.25 R. C. Suich, R. L. Patterson: *k*-out-of-*n*:G systems: some cost considerations., IEEE Trans. Reliab. **40**, 259–264 (1991)

54.26 D. H. Chi, W. Kuo: Reliability optimization of series–parallel systems using a genetic algorithm, Comp. Ind. Eng. **45**, 254–260 (1996)

54.27 B. F. Mitchell, R. J. Murry: Predicting operational availability for systems with redundant, repairable components and multiple sparing levels, Proc. Ann. Reliab. Maintainab. Symp., 301-305 (IEEE, 1996),

54.28 K. B. Misra, U. Sharma: An efficient algorithm to solve integer programming problems arising in system reliability design, IEEE Trans. Reliab. **R40**, 81–91 (1991)

54.29 K. B. Misra, V. Misra: A procedure for solving general integer programming problems, Microelectron. Reliab. **34**, 157–163 (1994)

54.30 V. R. Prasad, W. Kuo: Reliability optimization of coherent systems, IEEE Trans. Reliab. **49**, 323–330 (2000)

54.31 R. E. Barlow, F. Proschan: *Mathematical Theory of Reliability* (SIAM, Philadelphia 1996)

54.32 S. V. Amari, H. Pham: Optimal design of *k*-out-of-*n*:G subsystems subjected to imperfect fault-coverage, IEEE Trans. Reliab. **53**, 567–575 (2004)

54.33 S. V. Amari: Reliability, risk and fault-tolerance of complex systems. Ph.D. Thesis (Indian Institute of Technology, Kharagpur 1997)

54.34 M. Sasaki, S. Kaburaki, S. Yanagi: System availability and optimum spare units, IEEE Trans. Reliab. **R26**, 182–188 (1977)

54.35 G. L. Bilbro: Fast stochastic global optimization, IEEE Trans. Syst. Man Cyber. **24**, 684–689 (1994)

54.36 J. M. Renders, S. P. Flasse: Hybrid methods using genetic algorithms for global optimization, IEEE Trans. Syst. Man Cyber. B **26**, 243–258 (1996)

54.37 M. Zou, X. Zou: Global optimization: An auxiliary cost function approach, IEEE Trans. Syst. Man Cyber. A **30**, 347–354 (2000)

54.38 D. E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison Wesley, Reading 1989)

54.39 M. Gen, J. Kim: GA-based reliability design: state-of-the-art survey, Comp. Ind. Eng. **37**, 151–155 (1999)

54.40 I. O. Bohachevsky, M. E. Johnson, M. L. Stein: Generalized simulated annealing for function optimization, Technometrics **28**, 209–218 (1986)

54.41 B. Li, W. Jiang: A novel stochastic optimization algorithm, IEEE Trans. Syst. Man Cyber. B **30**, 193–198 (2000)

Acknowledgements

B.13 Uniform Design and Its Industrial Applications

by Kai-Tai Fang, Ling-Yau Chan

Kai-Tai Fang would like to express his gratitude for financial support from Hong Kong RGC grant RGC/HKBU 2044/02P and FRG grant FRG/03-04/II-711.

B.14 Cuscore Statistics: Directed Process Monitoring for Early Problem Detection

by Harriet B. Nembhard

This work was partially supported by NSF Grant #0451123.

C.19 Statistical Survival Analysis with Applications

by Chengjie Xiong, Kejun Zhu, Kai Yu

Dr. Xiong's work was partly supported by National Institute on Aging (USA) grants AG 03991 and AG 05681. Dr. Xiong's work and Prof. Zhu's work were also partly supported by the National Natural Science Foundation grant no. 70273044 of the People's Republic of China.

D.28 Measures of Influence and Sensitivity in Linear Regression

by Daniel Peña

This research has been supported by DGES projects SEJ 2004-03303, and CAM 06/HSE/0016/2004, Spain. I am very grateful to Juan Miguel Marín and Julia Villadomat for helpful comments.

D.29 Logistic Regression Tree Analysis

by Wei-Y. Loh

Research partially supported by grants from the National Science Foundation and the U.S. Army Research Office. The author thanks Dr. Kin-Yee Chan for codeveloping the LOTUS algorithm and for maintaining the software. The software may be obtained through a link on the author's website www.stat.wisc.edu/~loh.

D.32 Statistical Genetics for Genomic Data Analysis

by Jae K. Lee

This study was supported by the American Cancer Society grant RSG-02-182-01-MGO.

D.34 Statistical Methods In Proteomics

by Weichuan Yu, Baolin Wu, Tao Huang, Xiaoye Li, Kenneth Williams, Hongyu Zhao

This work was supported in part from NHLBI N01–HV-28186, NIGMS R01-59507, and NSF DMS 0241160.

E.39 Cluster Randomized Trials: Design and Analysis

by Mirjam Moerbeek

The research described in this chapter is partially funded by the Netherlands' Organization for Scientific Research (NWO), grant number 451-02-118.

E.40 A Two-Way Semilinear Model for Normalization and Analysis of Microarray Data

by Jian Huang, Cun-Hui Zhang

The research of Jian Huang is supported in part by the NIH grant HL72288-01 and an Iowa Informatics Initiative grant. The research of Cun-Hui Zhang is partially supported by the NSF grants DMS-0203086 and DMS-0405202. The authors thank Professor Terry Speed and his collaborators for making the Apo A1 data set available online.

E.41 Latent Variable Models for Longitudinal Data with Flexible Measurement Schedule

by Haiqun Lin

This chapter was written with partial support from NIMH grant R01 MH66187-01A2.

E.44 Condition-Based Failure Prediction

by Shang-K. Yang

This chapter quotes the contents of following papers with permission from Elsevier:

1. Yang, S. K. and Liu, T. S.: State estimation for predictive maintenance using Kalman filter, *Reliab. Eng. Sys. Saf.*, **66**, 29–39 (1999)
2. Yang, S. K.: An experiment of state estimation for predictive maintenance using Kalman filter on a DC motor, *Reliab. Eng. Sys. Saf.*, **75**, 103–111 (2002)

F.50 Six Sigma

by Fugee Tsung

The author thanks the HKUST Quality Lab student team for conducting an extensive review of Six Sigma for the input of this chapter. This work was supported by RGC Competitive Earmarked Research Grants HKUST6183/03E and HKUST6232/04E.

About the Authors

Susan L. Albin

Rutgers University
Department of Industrial and Systems Engineering
Piscataway, NJ, USA
salbin@rci.rutgers.edu

Chapter B.12

Susan L. Albin is professor and director of the Graduate Program of Industrial and Systems Engineering at Rutgers University. Dr. Albin's area of research is quality engineering, multivariate statistics, process control, and data mining. Her work has been applied in semiconductor manufacturing, plastics recycling, food processing, and medical devices and has been supported by NSF, FAA, DOD, and industrial organizations. Dr. Albin is secretary of INFORMS and Focus Issue editor for IIE Transactions on Quality and Reliability Engineering.

Suprasad V. Amari

Senior Reliability Engineer Relex Software Corporation
Greensburg, PA, USA
suprasad.amari@relex.com

Chapter F.54

Dr. Amari is a senior reliability engineer at Relex Software Corporation. He received both his M.S. and Ph.D. in reliability engineering from the Indian Institute of Technology, Kharagpur, India. He is an editorial board member of the International Journal of Reliability, Quality and Safety Engineering and International Journal on Performability Engineering, and management committee member of RAMS. He is an ASQ-certified reliability engineer and is a senior member of IEEE, IIE and ASQ.

Y. Alp Aslandogan

The University of Texas at Arlington
Computer Science and Engineering
Arlington, TX, USA
alp@cse.uta.edu
http://ranger.uta.edu/~alp

Chapter D.36

Dr. Aslandogan's main areas of research are biomedical informatics, data mining, multimedia information retrieval and visualization. He received his Ph.D. in computer science from the University of Illinois at Chicago in 2001. He has served and continues to serve on the technical program committees of IEEE International Conference on Information Technology and IEEE International Conference on Information Reuse and Integration. His recent research projects include a 3D change detection system for surface structures, a biomedical data mining web service and a concept-based multimedia search agent.

Jun Bai

JP Morgan Chase
Card Services
Wilmington, DE, USA
jun.bai@comcast.net
http://www.stat.rutgers.edu/~jbai

Chapter A.7

Dr. Jun Bai is a senior analyst at JP Morgan Chase Card Services. He obtained his Ph.D. in industrial and systems engineering from Rutgers – the State University of New Jersey in 2004. His research interests include warranty analysis, maintenance, reliability and applied statistics. Currently his research activities focus on risk management and statistical modelling in the financial industry.

Jaiwook Baik

Korea National Open University
Department of Information Statistics
Seoul, South Korea
baik@knou.ac.kr

Chapter A.5

Jaiwook Baik received the Ph.D. degree from the Department of Statistics, Virginia Polytechnic Institute and State University in 1991. Since 1992 he has been with the Department of Information Statistics, Korea National Open University, where he is currently a professor of the same department. His current research interests include warranty data analysis and applications of quality control techniques to solve industrial problems. He is a member of the Korean Reliability Society and the Korean Society for Quality Management.

Amit K. Bardhan

University of Delhi – South Campus
Department of Operational Research
New Delhi, India
amit@or.du.ac.in
http://people.du.ac.in/amit

Chapter C.25

Amit Kumar Bardhan is a senior lecturer in the Department of Operational Research, University of Delhi – South Campus. He obtained his Ph.D. in operational research from University of Delhi in 2003. His Ph.D. thesis was judged the best thesis in O.R. of the year by the Operational Research Society of India. His research interests are mathematical modelling, quality and reliability analysis and marketing models.

Anthony Bedford

Royal Melbourne Institute of Technology University
School of Mathematical and Geospatial Sciences
Bundoora, Victoria, Australia
anthony.bedford@rmit.edu.au
http://www.rmit.edu.au/math-geo

Chapter F.52

Dr. Bedford is a senior lecturer and researcher in statistics at Royal Melbourne Institute of Technology (RMIT) University. He completed his Ph.D. in 2003 on multi-priority finite buffer queueing models. His main areas of research are queueing theory in telecommunications systems, advances in medical statistics and sports statistics. He is also involved in postgraduate statistics research in occupation health and safety and the medical sciences.

James Broberg

Royal Melbourne Institute of Technology University
School of Computer Science & Information Technology
Melbourne, Victoria, Australia

Chapter F.52

James Broberg is currently a Ph.D. student working at Royal Melbourne Institute of Technology (RMIT) University, Melbourne (Australia). He is a member of the "Distributed Systems and Networking" discipline at RMIT, and has worked and published in the area of task assignment (e.g. scheduling policies) to enable effective load balancing and load sharing in distributed systems.

Michael Bulmer

University of Queensland
Department of Mathematics
Brisbane, Qld, Australia
m.bulmer@uq.edu.au
http://www.maths.uq.edu.au/~mrb

Chapter A.5

Dr. Bulmer is a senior lecturer in mathematics and statistics at the University of Queensland. He obtained his Ph.D. from the University of Tasmania in 1996 on the topic of automated algebraic reasoning. His current research interests include computational methods in statistics and operations research, stochastic modelling in astrophysics, and mathematics and statistics education.

Zhibin Cao

Arizona State University
Computer Science & Engineering Department
Tempe, AZ, USA
zbcao@asu.edu
http://www.public.asu.edu/~zcao2/

Chapter C.24

Mr. Zhibin Cao received his M.S. degree from Computer Science & Engineering Department at Arizona State University. Currently he is a Ph.D. candidate in the department. He worked at Peiking University Research and Development Institute, China and at Bell-Labs China before joining Arizona State University. His research areas include software engineering, service-oriented computing, service-oriented modelling and model-based development.

Philippe Castagliola

Université de Nantes and IRCCyN UMR CNRS 6597 Institut Universitaire de Technologie de Nantes
Qualité Logistique Industrielle et Organisation
Carquefou, France
philippe.castagliola@univ-nantes.fr
http://philippe.castagliola.free.fr/

Chapter B.17

Philippe Castagliola graduated (Ph.D. 1991) from the Université de Technologie de Compiègne, France (UTC). He is currently a professor at the IUT (Institut Universitaire de Technologie) de Nantes, France, and he is also a member of the IRCCyN (Institut de Recherche en Communications et Cybernétique de Nantes), UMR CNRS 6597. He is associate editor for the Journal of Quality Technology and Quantitative Management and for the International Journal of Reliability, Quality and Safety Engineering. His research activity includes developments of new SPC techniques (non-normal control charts, optimized EWMA type control charts, multivariate capability indices, and monitoring of batch processes).

Giovanni Celano

University of Catania
Dipartimento di Ingegneria Industriale e Meccanica
Catania, Italy
gcelano@diim.unict.it
http://www.diim.unict.it/users/gcelano/

Chapter B.17

Giovanni Celano received his Ph.D. in 2003 from the University of Palermo for work on the sequencing of mixed model assembly lines. He is currently assistant professor of technology and manufacturing systems at the University of Catania, Italy. His research is focused on statistical quality control and production scheduling. He is a member of the AITEM and of the ENBIS.

Ling-Yau Chan

The University of Hong Kong
Department of Industrial and Manufacturing Systems Engineering
Hong Kong
plychan@hku.hk
http://www.hku.hk/imse

Chapter B.13

Dr. Chan's research areas include design of industrial experiments, optimal design, uniform design, statistical quality control, reliability, maintenance, quality management, and supply chain management. He has published over 80 papers in these areas, and is collaborating with scholars worldwide on various research topics. He is the head of the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong.

Ted Chang

University of Virginia
Department of Statistics
Charlottesville, VA, USA
tcc8v@virginia.edu
http://www.stat.virginia.edu/chang.html

Chapter D.31

Ted Chang received his Ph.D. in mathematics from the University of California (Berkeley) in 1972. After about a decade of working in algebraic topology, he switched his research concentration to statistical problems in which geometry and symmetries play an important role. The primary applications of his work are in the estimation of the statistical errors in tectonic plate reconstructions, image reconstruction, and human motion data.

Victoria Chen

University of Texas at Arlington
Industrial and Manufacturing Systems Engineering
Arlington, TX, USA
vchen@uta.edu
http://ie.uta.edu/index.cfm?fuseaction=professordescription&userid=1945

Chapter D.36

Dr. Victoria Chen joined the University of Texas at Arlington in 2001. From 1993–2001 she was on the Industrial and Systems Engineering faculty at the Georgia Institute of Technology. She earned her Ph.D. in operations research and industrial engineering from Cornell University. Dr. Chen is co-founder of the Informs Section on Data Mining and is currently serving as chair.

Yinong Chen

Arizona State University
Computer Science and Engineering Department
Tempe, AZ, USA
yinong.chen@asu.edu
http://www.public.asu.edu/~ychen10/

Chapter C.24

Dr. Yinong Chen received his Ph.D. from the University of Karlsruhe, Germany. He worked at LAAS-CNRS, France, and at Wits University, South Africa, before joining Arizona State University. His research areas include dependable computing, software engineering, and service-oriented computing. He has coauthored five books and over 80 research papers in these areas.

Peter Dimopoulos

Royal Melbourne Institute of Technology University
Computer Science and IT
Melbourne, Australia
dimpet@cs.rmit.edu.au
http://www.cs.rmit.edu.au/~dimpet

Chapter F.52

Peter is currently completing his Ph.D. in computer science at RMIT University (Royal Melbourne Institute of Technology) in the area of Internet congestion control. Prior to his Ph.D. he worked at Agilent Technologies and completed a double degree in computer science and computer systems engineering at RMIT University.

Fenghai Duan

Department of Preventive and Societal Medicine
Omaha, NE, USA
fduan@unmc.edu

Chapter D.33

Fenghai Duan was born in Heilongjiang, China. After completing his bachelor's degree in biochemistry at Fudan University in 1995, he received his master's degree in molecular biology from Institute of Biophysics, Academia Sinica in 1998. In the year 2000, Duan joined the Ph.D. program in biostatistics at Yale University and worked on his thesis in the lab of Professor Heping Zhang. Duan's doctoral dissertation was about the analysis of microarray experiments and was awarded the Ph.D. degree in May 2005. Currently, he is an assistant professor of the Department of Preventive and Societal Medicine at University of Nebraska Medical Center. His research interests are in the development of statistical methods for the analysis of high-dimensional biological data.

Veronica Esaulova

Otto-von-Guericke-University
Magdeburg
Department of Mathematics
Magdeburg, Germany
veronica.esaulova@gmail.com

Chapter C.20

Miss Esaulova has submitted her Ph.D. dissertation devoted to hazard rate modeling in heterogeneous populations and about to defend it in June 2006. She has publications in the fields of survival analysis and nonparametric statistics and is interested in the development of statistical methodology and its applications.

Luis A. Escobar

Louisiana State University
Department of Experimental Statistics
Baton Rouge, LA, USA
luis@lsu.edu
http://www.stat.lsu.edu/faculty/Escobar

Chapter C.22

Luis A. Escobar is a professor at Louisiana State University. His research interests include analysis of reliability data and accelerated testing. Luis is an associate editor for LIDA and past associate editor for Technometrics. He is a Fellow of the ASA and elected member of the ISI. Luis was awarded the 1999 Jack Youden Prize.

Chun Fan

Arizona State University
Computer Science & Engineering Department
Tempe, AZ, USA
fanchun@asu.edu
http://whoknows.eas.asu.edu/~whoknows/

Chapter C.24

Mr. Chun Fan is a Ph.D. student in the Department of Computer Science and Engineering at Arizona State University. He received his B.E. degree from the University of Science and Technology of China, China. His research areas include software engineering, software architecture, and computer-based simulation.

Kai-Tai Fang

Hong Kong Baptist University
Department of Mathematics
Kowloon Tong, Hong Kong
ktfang@hkbu.edu.hk
http://www.math.hkbu.edu.hk/~ktfang

Chapter B.13

Professor Fang is an elected fellow of the Institute of Mathematical Statistics and of the American Statistical Association. He was chair professor at the Hong Kong Baptist University from 1993 to 2006. His research interest: involve multivariate analysis, experimental design, data mining and statistical inference. He has been associate editor for Statistics & Probability Letters, Statistica Sinica and Journal of Multivariate Analysis. He has published more than 220 research papers and 18 books.

Qianmei Feng

University of Houston
Department of Industrial Engineering
Houston, TX, USA
qfeng@central.uh.edu

Chapter B.11

Dr. Qianmei Feng is an assistant professor in the Department of Industrial Engineering at the University of Houston, TX. Her research interests are quality and reliability engineering, especially inspection strategies, tolerance design and optimization, experimental design, and Six Sigma. She is a member of IIE, INFORMS and Alpha Pi Mu.

Emilio Ferrari

University of Bologna
Department of Industrial and Mechanical Engineering (D.I.E.M.)
Bologna, Italy
emilio.ferrari@unibo.it

Chapter F.48

Emilio Ferrari is full professor of industrial logistics at the Department of Mechanical Constructions (DIEM) at the University of Bologna, director of the master degree in "integrated logistics" at the Faculty of Engineering in Bologna and of the Summer School "Francesco Turco" on industrial plants. He is author of more than 70 publications, most of them about industrial logistics and industrial plant design and management.

Sergio Fichera

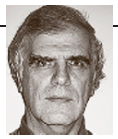
University of Catania
Department Industrial and Mechanical Engineering
Catania, Italy
sfichera@diim.unict.it

Chapter B.17

Sergio Fichera is an associate professor of Technology and Manufacturing System at the Dipartimento di Ingegneria Industriale eMeccanica of the University of Catania, Italy. He holds an M.B.A. degree from the Schools of Management at the University of Turin. His research interests are in production scheduling and statistical quality control. He is a member of the AITEM (Italian association for manufacturing).

Maxim Finkelstein

University of the Free State
Department of Mathematical Statistics
Bloemfontein, South Africa
finkelm.sci@mail.uovs.ac.za
www.uovs.ac.za/departments/mathstats/finkelsteinms



Chapter C.20

Professor Maxim Finkelstein is a specialist in reliability theory and other applications of stochastic modeling. He has published about 140 papers and 4 books on various aspects of reliability and survival analysis. His current major interest is in stochastic modeling of heterogeneity for engineering and biological applications. He is a member of editorial boards of a number of reliability oriented journals.

Mitsuo Gen

Waseda University
Graduate School of Information, Production & Systems
Kitakyushu, Japan
gen@waseda.jp

Chapter E.42

Mitsuo Gen received his Ph.D. degree from Kogakuin University, Japan, in 1974. He is a professor at the Graduate School of Information, Production and Systems, Waseda University. He was a visiting professor at the University of California, Berkeley in 1999–2000. His research interests include genetic algorithms, neural networks, fuzzy logic, and the applications to network design, scheduling, system reliability design, and the like.

Amrit L. Goel

Syracuse University
Department of Electrical Engineering and Computer Science
Syracuse, NY, USA
goel@ecs.syr.edu

Chapters D.35, F.53

Amrit L. Goel is a professor of EECS at Syracuse University. His Ph.D. was in mechanical engineering from the University of Wisconsin, Madison. His academic activities have included quality control and reliability, software engineering, databases, and data mining using radial basis functions (RBF) and support vector machines. He has advised fifteen Ph.D. dissertations on these and related topics. He is a co-author of a book on object-oriented software testing with Dr. Bashir, of the Goel–Okumoto software reliability model and, most recently, of the Shin–Goel algorithm for RBF design. He was elected a fellow of IEEE for contributions to software reliability.

Thong N. Goh

National University of Singapore
Industrial and Systems Engineering Dept.
Singapore, Republic of Singapore
isegohtn@nus.edu.sg
http://www.ise.nus.edu.sg/staff/gohtn

Chapter B.16

Professor Thong N. Goh (Ph.D. University of Wisconsin – Madison, USA) is academician of the International Academy for Quality, fellow of the American Society for Quality, and IEEE Engineering Management Society Educator of the Year 2004. Specializing in statistical methodologies for engineering applications, he now serves on the editorial boards of several leading international research journals on quality management and quality engineering.

Raj K. Govindaraju

Massey University
Institute of Information Sciences
and Technology
Palmerston North, New Zealand
k.govindaraju@massey.ac.nz

Chapter B.15

Dr. Govindaraju holds a Ph.D. degree from Madras University and has been engaged in statistics teaching and consulting for the last 20 years. His research interest is in the statistical aspects of quality and data analysis. He is an associate editor of the International Journal for Economic Quality.

Xuming He

University of Illinois at
Urbana-Champaign
Department of Statistics
Champaign, IL, USA
x-he@uiuc.edu
http://www.stat.uiuc.edu/~x-he

Chapter D.30

Professor He's research focuses on statistical inference for regression models. He is an elected fellow of the Institute of Mathematical Statistics, and currently serves on the editorial boards of The Annals of Statistics and Journal of the American Statistical Association. His research has been supported by the NSF, NSA and NIH.

Chengcheng Hu

Harvard School of Public Health
Department of Biostatistics
Boston, MA, USA
cchu@hsph.harvard.edu
http://www.hsph.harvard.edu/faculty/ChengchengHu.html

Chapter C.21

Dr. Chengcheng Hu is an assistant professor of biostatistics at the Harvard School of Public Health and a senior statistician at the Statistical and Data Analysis Center of the Pediatric AIDS Clinical Trials Group. He earned his Ph.D. in biostatistics from the University of Washington in 2001. His research interests are in the areas of failure time data analysis, measurement error, missing data, longitudinal data analysis, and design of clinical trials.

Feifang Hu

University of Virginia
Department of Statistics
Charlottesville, VA, USA
fh6e@virginia.edu
http://www.stat.virginia.edu/hu.html

Chapter E.37

Dr. Hu is an associate professor of Statistics at the University of Virginia. He has a Ph. D. from the University of British Columbia and has worked at the National University of Singapore and Cornell University. His main research areas are: bootstrap methods; biostatistics; likelihood inference and data mining.

Hai Huang

Intel Corp CH3-20
Component Automation Systems
Chandler, AZ, USA
hai.huang2@intel.com

Chapter C.24

Dr. Hai Huang received his Ph.D. from the Arizona State University, USA. Currently he works in the Component Automation Systems (CAS) Group at Intel Corp. His research areas include software verification and validation, test automation, Web services, service-oriented architecture, and compiler technology.

Jian Huang

University of Iowa
Department of Statistics and Actuarial
Science
Iowa City, IA, USA
jian-huang@uiowa.edu
http://www.stat.uiowa.edu/~jian/

Chapter E.40

Dr. Jian Huang obtained his Ph.D. in statistics from The University of Washington in Seattle. His current research interests include statistical analysis of high-dimensional data with applications to biomedical research, statistical genetics, survival analysis, and semiparametric models.

Tao Huang

Yale University, School of Medicine
Department of Epidemiology and Public Health
New Haven, CT, USA
t.huang@yale.edu

Chapter D.34

Dr. Tao Huang is a postdoctoral associate in Department of Epidemiology and Public Health at Yale University School of Medicine. He obtained his Ph.D. degree in statistics from the University of North Carolina at Chapel Hill. His current research interests include nonparametric and semiparametric modeling, functional and longitudinal data analysis, model selection and computational biology and statistical genetics.

Wei Jiang

Stevens Institute of Technology
Department of Systems Engineering and Engineering Management
Hoboken, NJ, USA
wjiang@stevens.edu
http://www.stevens.edu/engineering/seem/People/jiang.html

Chapters B.10, D.36

Dr. Wei Jiang is an assistant professor of systems engineering and engineering management at Stevens Institute of Technology. He obtained his Ph.D. degree in industrial engineering and engineering management from Hong Kong University of Science and Technology in 2000. Prior to joining Stevens, he worked as a statistical consultantat AT&T Labs, Morristown. His current research activities include statistical methods for quality control, data mining and enterprise intelligence.

Richard Johnson

University of Wisconsin – Madison
Department of Statistics
Madison, WI, USA
rich@stat.wisc.edu

Chapter B.18

Richard A. Johnson is a professor of statistics at the University of Wisconsin. His research and consulting interests include reliability and life length analysis, applied multivariate analysis, and applications to engineering. He is a fellow of the American Statistical Association, fellow of the Institute of Mathematical Statistics, and elected member of the International Statistical Institute. He has been editor of Statistics and Probability Letters since it began 25 years ago, and is co-author of six books and several book chapters, and over 100 papers in the statistical and engineering literature.

Kailash C. Kapur

University of Washington
Industrial Engineering
Seattle, WA, USA
kkapur@u.washington.edu
http://faculty.washington.edu/kkapur/

Chapter B.11

Dr. Kailash C. Kapur is a professor of industrial engineering at the University of Washington. He received Ph.D. degree (1969) in industrial engineering from the University of California at Berkeley. He received the Allan Chop Technical Advancement Award and the Craig Award from ASQ. He is a Fellow of ASQ and IIE, and a registered professional engineer. In his present position at the University of Washington, Dr. Kapur is responsible for teaching and research in the areas of quality engineering, design reliability, industrial experimental design, system optimization and control, and productivity improvement.

P. K. Kapur

University of Delhi
Department of Operational Research
Delhi, India
pkkapur@or.du.ac.in

Chapter C.25

P.K. Kapur is professor and head in the Department of Operational Research, University of Delhi. He obtained his Ph.D. degree from the University of Delhi in 1977. He has published extensively in Indian journals and abroad in the areas of hardware reliability, optimization, queueing theory, and maintenance and software reliability. He is currently the president of the Operational Research Society of India.

Kyungmee O. Kim

Konkuk University
Department of Industrial Engineering
Seoul, S. Korea
kyungmee@konkuk.ac.kr
http://mail.konkuk.ac.kr/~kyungmee

Chapter A.9

Dr. Kim is an assistant professor of industrial engineering at Konkuk University in Seoul, Korea. She obtained her Ph.D. degree in the Department of Industrial Engineering from Texas A&M in 1999. Her research interests include statistical quality control, burn-in, yield and reliability optimization, fault diagnosis and condition-based maintenance.

Taeho Kim

Korea Telecom
Strategic Planning Office
Sungnam, Kyonggi-do, S. Korea
taehokim@kt.co.kr

Chapter A.9

Taeho Kim received his Ph.D. from Texas A&M University in 1998. He is assistant vice-president of Korea Telecom. Since he joined KT in 1986 he has been doing many projects related with service quality, facility reliability, and network performance. His current fields of interest include six sigma for continuous growth and quality innovation in the telecom industry.

Way Kuo

University of Tennessee
Department of Electrical and Computer Engineering
Knoxville, TN, USA
way@utk.edu
http://www.ece.utk.edu/bios/Faculty/Kuo.html

Chapter A.9

Dr. Way Kuo is university distinguished professor and dean of engineering at the University of Tennessee. He is an elected member of the US National Academy of Engineering, Academia Sinica, Taiwan, R.O.C., and the International Academy for Quality. He has co-authored five textbooks and currently serves as the editor of IEEE Transactions on Reliability. He is recognized as one of the principal scholars responsible for developing cost-effective methodologies for reducing infant mortality in the fast-evolving microelectronics industry. His contributions to industry include advancing the development of the fundamentals of reliability design as well as introducing new industrial applications of parametric and nonparametric analysis.

Paul Kvam

Georgia Institute of Technology
School of Industrial and Systems Engineering
Atlanta, GA, USA
pkvam@isye.gatech.edu
http://www.isye.gatech.edu/~pkvam/

Chapter A.2

Paul Kvam is an associate professor in the School of Industrial and Systems Engineering (ISyE). He joined ISyE in 1995 after working for four years as scientific staff at the Los Alamos National Laboratory. Dr. Kvam received his B.S. in mathematics from Iowa State University in 1984, an M.S. in statistics from the University of Florida in 1986, and his Ph.D. in statistics from the University of California, Davis in 1991. His research interests focus on statistical reliability with applications to engineering, nonparametric estimation, and analysis of complex and dependent systems. He is a member of the American Statistical Association, Institute of Mathematical Statistics, Institute for Operations Research and Management Science, and IEEE.

Chin-Diew Lai

Massey University
Institute of Information Sciences and Technology
Palmerston North, New Zealand
c.lai@massey.ac.nz
http://www-ist.massey.ac.nz/ResearchGroups/DisplayStaff.asp?StaffID=24

Chapter A.3

Chin-Diew Lai received his Ph.D. in statistics from Victoria University of Wellington, New Zealand in 1975. His main research interests are in quality and reliability engineering. He is a co-author of three books and has published over 90 journal articles and book chapters.

Jae K. Lee

University of Virginia
Public Health Sciences
Charlottesville, VA, USA
jaeklee@virginia.edu
http://www.healthsystem.virginia.edu/internet/hes/personnel/leejk.cfm

Chapter D.32

Professor Lee received his Ph.D. in statistical genetics from the University of Wisconsin – Madison in 1995. He has worked on statistical research in molecular genetics and bioinformatics, including genetic population inference, DNA structure analysis, linkage association study, and high-throughput gene chip data analysis on various biomedical studies. In particular, he has pioneered the statistical development of small-sample microarray data analysis techniques such as LPE (local pooled error) and HEM (heterogeneous error model) for practical, genomic biomedical investigations.

Kit-Nam F. Leung

City University of Hong Kong
Department of Management Sciences
Kowloon Tong, Hong Kong
msknleun@cityu.edu.hk

Chapter F.49

Dr. Kit-Nam Francis Leung received a B.Sc. degree in mathematics in 1984 and M.Sc. degree in operational research in 1985, both from London University, and his Ph.D. in 2003 from Curtin University, Australia. Since 1988 he has been a lecturer in the Management Sciences Department at the City University of Hong Kong and has been responsible for teaching management science and Statistics. His research interests are maintenance, reliability and warranty.

Ruojia Li

Global Statistical Sciences
Lilly Corporate Center
Indianapolis, IN, USA
liru@lilly.com

Chapter B.18

Dr. Ruojia Li is a research scientist at Eli Lilly and Company. She received her Ph.D. degree in statistics from the University of Wisconsin – Madison in 2004. Her research interests include multivariate quality monitoring schemes and statistical applications in pharmaceutical research.

Wenjian Li

Javelin Direct, Inc.
Marketing Science
Irving, TX, USA
wenjian.li@javelindirect.com

Chapter E.45

Dr. Wenjian Li is a marketing research scientist with Javelin Direct, Inc., whose current work focuses on econometrics, forecasting, and survival analysis. Dr. Li earned his Ph.D. from Rutgers University where his primary research interests included reliability, maintenance theory, applied statistics and manufacturing automation.

Xiaoye Li

Yale University
Department of Applied Mathematics
New Heaven, CT, USA
xiaoye.li@yale.edu

Chapter D.34

Xiaoye Li is currently a Ph.D. candidate in the Applied Mathematics Department of Yale University. He is interested in machine learning and statistical learning theory and the application of machine learning techniques to various data mining problems, especially those arising from genomics and proteomics studies. His dissertation takes an initiative step to analyze the popular classification algorithm Random Forest and to improve the random subspace method.

Yi Li

Harvard University
Department of Biostatistics
Boston, MA, USA
yili@jimmy.harvard.edu
http://www.hsph.harvard.edu/faculty/YiLi.html

Chapter E.38

Dr. Li is an associate professor of biostatistics at the University of Cincinnati. He obtained his Ph.D. degree in biostatistics from the University of Michigan in 1999. He has been working on survival analysis, longitudinal/spatial data analysis and observational studies, He is the recipient of several prestigious awards, including the David P. Byar Young Investigator Award, John van Ryzin Award and Roger L. Nichols Excellence in Teaching Award. He is a member of the review panel of Mathematical Reviews and is an associate editor of Biometrics.

Hojung Lim

Korea Electronics Technology Institute (KETI)
Ubiquitous Computing Research Center
Seongnam-Si, Gyeonggi-Do, Korea
hlim@keti.re.kr

Chapter F.53

Hojung Lim received her Ph.D. in computer and information science from Syracuse University, New York. Her research interests are in support vector machines and software modelling. Currently she is involved in ubiquitous sensor networks and radio frequency identification (RFID).

Haiqun Lin

Yale University School of Medicine
Department of Epidemiology and Public Health
New Haven, CT, USA
haiqun.lin@yale.edu
http://publichealth.yale.edu/faculty/lin.htm

Chapter E.41

Haiqun Lin received her Ph.D. in biometry from Cornell University. She also holds a medical degree from Beijing Medical University, China. Haiqun Lin's current research focuses on latent variable modelling and missing data issues in longitudinal data. In the last a few years, Haiqun Lin has been collaborating with scientific researchers in the fields of cancer research, psychiatry, and geriatric medicine.

Nan Lin

Washington University in Saint Louis
Department of Mathematics
St. Louis, MO, USA
nlin@math.wustl.edu
http://www.math.wustl.edu/~nlin

Chapter D.30

Dr. Lin received his Ph.D. degree in statistics from the University of Illinois at Urbana-Champaign in 2003. He is an assistant professor in the Department of Mathematics, Washington University in Saint Louis. His research interest includes robust statistics, Bayesian modeling, and applications of statistical methodologies in bioinformatics studies such as protein–protein interaction prediction and topological structure inference in yeast.

Wei-Yin Loh

University of Wisconsin – Madison
Department of Statistics
Madison, WI, USA
loh@stat.wisc.edu
http://www.stat.wisc.edu/~loh

Chapter D.29

Wei-Yin Loh has a Ph.D. from Berkeley. He is a fellow of the American Statistical Association and the Institute of Mathematical Statistics. He invented the GUIDE regression tree algorithm and co-authored the CRUISE, LOTUS, and QUEST algorithms. He currently serves on the editorial boards of the ACM Transactions on Knowledge Discovery from Data and the Journal of Machine Learning Research.

Jye-Chyi Lu

The School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA, USA
jclu@isye.gatech.edu
http://www.isye.gatech.edu/~JCLU;
http://www.isye.gatech.edu/faculty-staff/profile.php?entry=jl234

Chapter A.2

Jye-Chyi (JC) Lu is a professor in the School of Industrial and Systems Engineering (ISyE). He received a Ph.D. in statistics from University of Wisconsin at Madison in 1988, and joined the faculty of North Carolina State University (NCSU) where he remained until 1999 when he joined ISyE. Dr. Jye-Chyi Lu's research areas cover industrial statistics, signal processing, semiconductor and electronic manufacturing, data mining, bioinformatics, supply-chain management, logistics planning and nanotechnology. He has about 58 disciplinary and interdisciplinary publications, which have appeared in both engineering and statistics journals. Currently, he is an associate editor for Technometrics, IEEE Transactions on Reliability and Journal of Quality Technology.

William Q. Meeker, Jr.

Iowa State University
Department of Statistics
Ames, IA, USA
wqmeeker@iastate.edu
http://www.public.iastate.edu/~wqmeeker

Chapter C.22

Dr. Meeker is distinguished professor of statistics at Iowa State University. He is a fellow of the American Statistical Association and a past editor of Technometrics. He is co-author of the books *Statistical Methods for Reliability Data*, and *Statistical Intervals*, and of numerous publications in the engineering and statistical literature. He has consulted extensively on problems in reliability and accelerated testing.

Mirjam Moerbeek

Utrecht University
Department of Methodology and Statistics
Utrecht, Netherlands
m.moerbeek@fss.uu.nl
http://www.fss.uu.nl/ms/moerbeek

Chapter E.39

Dr. Mirjam Moerbeek studied biometrics at Wageningen Agricultural University, the Netherlands. She obtained her Ph.D. from Maastricht University, the Netherlands. She is currently employed at Utrecht University, the Netherlands. In 2003 she received a prestigious research grant from the Dutch government for young researchers. Her research topic is on the design and analysis of experiments with nested data.

Terrence E. Murphy

Yale University School of Medicine
Department of Internal Medicine
New Haven, CT, USA
terrence.murphy@yale.edu

Chapter B.10

Terrence E. Murphy earned his Ph.D. in industrial and systems engineering from the Georgia Institute of Technology in 2004. Prior to his graduate work in engineering statistics, he worked for the Eastman Kodak and Johnson & Johnson companies in the manufacture and development of clinical instrumentation. His interests include multivariate statistics, experimental design and medical decision making.

D.N. Pra Murthy

The University of Queensland
Division of Mechanical Engineering
Brisbane, QLD, Australia
p.murthy@uq.edu.au

Chapters A.3, A.5

Pra Murthy obtained his Ph.D. degree from Harvard University. He has authored or co-authored 5 books, 20 book chapters and 150 journal papers and co-edited 2 books. His current areas of research deal with various topics in product reliability and product warranty. He has held visiting appointments at several universities in the USA, Europe and Asia and is on the editorial boards of nine international journals.

H. N. Nagaraja

Ohio State University
Department of Statistics
Columbus, OH, USA
hnn@stat.ohio-state.edu
http://www.stat.ohio-state.edu/~hnn

Chapter A.4

H.N. Nagaraja, Ph.D., is a professor in the Departments of Statistics and Internal Medicine and serves as a General Clinical Research Centre Biostatistician at Ohio State University. He is interested in order and record statistics, general distribution theory, stochastic modelling, and biostatistical applications. He is a fellow of the American Statistical Association, and an elected member of the International Statistical Institute.

Toshio Nakagawa

Aichi Institute of Technology
Department of Marketing and
Information Systems
Toyota, Japan
toshi-nakagawa@aitech.ac.jp
http://www.aitech.ac.jp/

Chapter E.46

Toshio Nakagawa received his Ph.D. from Kyoto University in 1977. He is now a professor of marketing and information systems at Aichi Institute of Technology in Toyota. His research interests are optimization problems, and computer and information systems in reliability and maintenance theory.

Joseph Naus

Rutgers University
Department of Statistics
Piscataway, NJ, USA
naus@stat.rutgers.edu
http://www.stat.rutgers.edu/people/faculty/naus.html

Chapter E.43

Joseph Naus is a professor of statistics at Rutgers University. He received his Ph.D in statistics from Harvard University. He was elected a Fellow of The American Statistical Association based on his research into scan statistics, a continuing research area for more than 40 years.

Harriet B. Nembhard

Pennsylvania State University
Harold and Inge Marcus Department of
Industrial and Manufacturing Engineering
University Park, PA, USA
hbn2@psu.edu
http://www.ie.psu.edu/People/IEFaculty/facultypage.cfm?FacID=18

Chapter B.14

Dr. Nembhard's research mission is to investigate the design and implementation of concepts and methods of quality, economics, productivity, and improvement for manufacturing systems. She is also an ASQ certified Six Sigma Black Belt and has served as an expert consultant for several major companies. She is on the editorial boards of Quality Engineering and Quality and Reliability Engineering International.

Douglas Noe

University of Illinois
at Urbana-Champaign
Department of Statistics
Champaign, IL, USA
dnoe@uiuc.edu

Chapter D.30

Douglas Noe is a Ph.D. candidate in the Department of Statistics at the University of Illinois at Urbana-Champaign. He earned an M.S. from this department in 2003 and received his M.A. in economics from the University of Michigan in 2000. His research explores statistical aspects of data mining.

Arrigo Pareschi

University of Bologna
Department of Industrial and Mechanical Engineering (D.I.E.M.)
Bologna, Italy
arrigo.pareschi@unibo.it

Chapter F.48

Arrigo Pareschi is is full professor of industrial logistics at the Department of Mechanical Constructions (DIEM) of the University of Bologna. He has been dean of Faculty of Engineering of Bologna from 1955 to 2001 and president of the Commission for Scientific Research and of the "Spin-Off" Committee of the University of Bologna. He is author of over 90 scientific papers (both experimental and theoretical) on industrial mechanical plants.

Francis Pascual

Washington State University
Department of Mathematics
Pullman, WA, USA
jpascual@math.wsu.edu

Chapter C.22

Dr. Francis Pascual received his Ph.D. in statistics from Iowa State University. He has a joint appointment in the Department of Statistics and the Department of Mathematics at Washington State University. His research interests include statistical analysis of reliability data, accelerated life test planning, statistical process control, and analysis of spatial correlations.

Raymond A. Paul

C2 Policy
U.S. Department of Defense (DoD)
Washington, DC, USA
raymond.paul@osd.mil

Chapter C.24

Ray Paul has been a professional electronics engineer, software architect, developer, tester and evaluator for the past 24 years, holding numerous positions in the field of software engineering. Currently, he serves as the deputy for C2 Metrics and Performance Measures for Software for the Department of Defense (DoD) Chief Information Officer (CIO). In this position, he supervises development of objective, quantitative data on the status of software resources in DoD information technology (IT) to support major investment decisions. These metric data are required to meet various congressional mandates, most notably the Clinger-Cohen Act. He holds a doctorate in software engineering and is an active member of the IEEE Computer Society. He has published more than 50 articles on software engineering in various technical journals and symposia proceedings, primarily under IEEE sponsorship.

Alessandro Persona

University of Padua
Department of Industrial and Technology Management
Vicenza, Italy
alessandro.persona@unipd.it

Chapter F.48

Alessandro Persona is a full professor of industrial plants and logistics in the Department of Management and Technology at Padua University. The scientific activity has been carried out in many areas of research in industrial plants, logistic and maintenance topics. He is author of more than 90 publications. In 2005 he received the award for the best paper printed in the Int. Journal of Manufacturing Technology Management. He is member of the editorial board of the International Journal on Operational Research. Currently he manages the Ph.D. on Mechatronics and Industrial Systems and he is the president of mechanical engineering degree at Padua University.

Daniel Peña

Universidad Carlos III de Madrid
Departamento de Estadistica
Getafe (Madrid), Spain
daniel.pena@uc3m.es
http://halweb.uc3m.es/daniel_pena

Chapter D.28

Daniel Peña is professor of statistics at the Universidad Carlos III of Madrid. He was full professor of statistics at Universidad Politécnica de Madrid and visiting full professor at the Universities of Wisconsin – Madison and Chicago. He has published 13 books and more than 150 research papers on time series, linear models, robust and diagnostic methods, bayesian statistics, econometrics, multivariate analysis and quality methods. He is a member of ISI and IMS fellow.

Hoang Pham

Rutgers University
Department of Industrial and Systems Engineering
Piscataway, NJ, USA
hopham@rci.rutgers.edu

Chapters A.1, A.7, C.27, E.45

Dr. Hoang Pham is professor in the Department of Industrial and Systems Engineering at Rutgers University. Before joining Rutgers, he was a senior engineering specialist at the Boeing Company, Seattle, and the Idaho National Engineering Laboratory, Idaho Falls. His research interests include software reliability, system reliability modeling, maintenance, fault-tolerant computing, and biological systemability-risk assessment. He is the author/editor of more than 15 books and is currently the editor of the Springer Series in Reliability Engineering. He has published more than 90 journal articles and 30 book chapters. Dr. Hoang Pham is a fellow of the IEEE.

John Quigley

University of Strathclyde
Department of Management Science
Glasgow, Scotland
j.quigley@strath.ac.uk
http://www.managementscience.org/staff/john.asp

Chapter A.6

Dr. John Quigley earned a BMath in actuarial science from the University of Waterloo, Canada and a Ph.D. from the Department of Management Science, University of Strathclyde, Scotland. His research interests include applied probability modelling, statistical inference and reliability growth modelling. He is a member of the Safety and Reliability Society, a chartered statistician and an associate of the Society of Actuaries.

Alberto Regattieri

Bologna University
Department of Industrial and Mechanical Engineering
Bologna, Italy
alberto.regattieri@mail.ing.unibo.it

Chapter F.48

Alberto Regattieri is a professor in the Department of Industrial and Mechanical Engineering at the University of Bologna. He received his Ph.D. degree from Parma University in 1999. His current research interests include the optimal design of manufacturing systems, production planning and control, and the maintenance of industrial plants. In 2005 he received the Williamson Award [Emerald Literati Club (UK)] for his studies. He has authored or co-authored several books and over 50 technical publications.

Miyoung Shin

Kyungpook National University
School of Electrical Engineering and Computer Science
Daegu, Republic of Korea
shinmy@knu.ac.kr

Chapter D.35

Dr. Miyoung Shin is an assistant professor in the School of Electrical Engineering and Computer Science at Kyungpook National University. She earned her Ph. D. degree in computer and information science from Syracuse University in 1998 and was awarded the All-University Doctoral Prize for her outstanding Ph.D. thesis. Prior to joining to Kyungpook National University in 2005, she had worked as a senior member of research staff in the Electronics and Communications Research Institute. Her current research interests include data mining algorithms, bioinformatics and context-awareness computing.

Karl Sigman

Columbia University in the City of New York,
School of Engineering and Applied Science
Center for Applied Probability (CAP)
New York, NY, USA
karl.sigman@columbia.edu
http://www.columbia.edu/~ks20

Chapter A.8

Professor Sigman's areas of research include stochastic modeling, stochastic networks and queueing theory, point processtheory, and insurance risk. He was a recipient of the Presidential Young Investigator Award from the National Science Foundation, and continues to be co-director of Columbia's Center for Applied Probability.

Loon C. Tang

National University of Singapore
Department of Industrial and Systems Engineering
Singapore, Singapore
isetlc@nus.edu.sg
http://www.ise.nus.edu.sg/staff/tanglc/index.html

Chapter C.23

Dr. Loon Ching Tang, a faculty member of National University of Singapore, obtained a Ph.D. degree in 1992 from Cornell University in the field of operations research. He has published more than 50 papers in international journals in the field of quality, reliability and operations research. In particular, his research interest lies in the application of probability, statistics and optimization techniques in solving real world problems. He is currently the area editor of the International Journal of Performability Engineering.

Charles S. Tapiero

Polytechnic University
Technology Management and Financial Engineering
Brooklyn, NY, USA
ctapiero@poly.edu

Chapter F.47

Charles S. Tapiero is the Topfer Chair Professor of Financial Engineering and Technology Management at the Polytechnic University of New York. He has a worldwide reputation as an active researcher and consultant in risk and computational finance and risk management. He is currently the area editor for finance in the Journal of Applied Stochastic Models for Business and Industry as well as a member of the editorial board of several other journals. Professor Tapiero has published 12 books and over 250 papers on a broad range of issues spanning risk management, stochastic modeling and applied stochastic control in operations, insurance and finance.

Zahir Tari

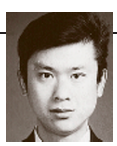
Royal Melbourne Institute of Technology University
School of Computer Science and Information Technology
Melbourne, Victoria, Australia
zahirt@cs.rmit.edu.au
http://www.cs.rmit.edu.au/~zahirt

Chapter F.52

Dr. Zahir Tari is a full professor at RMIT University and the director of Distributed Systems and Networking at the School of Computer Science and Information Technology. He has extensively published in the area of middlewares and Web services, especially in the area of performance (caching and load balancing), security (i.e. access control and information flow control) and service discovery.

Xiaolin Teng

Time Warner Inc.
Research Department
New York, NY, USA
xiaolin_teng@timeinc.com



Chapter C.27

Xiaolin Teng received his Ph.D. in industrial engineering from Rutgers University in 2001. He also holds master degrees in statistics, computer science, and automation. He is a member of ASA, INFORMS, IEEE and IIE. Currently Dr. Teng works at Time Warner Inc. as a research manager. His research interests include reliability, quality control, inventory optimization and data mining.

Wei-Tek Tsai

Arizona State University
Computer Science & Engineering
Department
Tempe, AZ, USA
wtsai@asu.edu
http://cse.asu.edu/directory/faculty/tsai.php

Chapter C.24

Professor Tsai received his Ph.D. from University of California at Berkeley 1985 and is professor of Computer Science and Science at Arizona State University. His research areas include service-oriented computing, software engineering, dependable computing, software engineering, software testing, and embedded systems. He has coauthored more than 300 research papers in these areas.

Kwok-Leung Tsui

Georgia Institute of Technology
School of Industrial and Systems Engineering
Atlanta, GA, USA
ktsui@isye.gatech.edu
http://www.isye.gatech.edu/~ktsui

Chapters B.10, D.36

Kwok-Leung Tsui is professor at Georgia Institute of Technology. He has a Ph.D. in statistics from the University of Wisconsin. Dr. Tsui is a (elected) fellow of American Statistical Association and was a recipient of the NSF Young Investigator Award. He is currently the departmental editor of the IIE Transactions.

Fugee Tsung

Hong Kong University of Science and Technology
Department of Industrial Engineering and Logistics Management
Kowloon, Hong Kong
season@ust.hk

Chapter F.50

Dr. Fugee Tsung is an associate professor in the Department of Industrial Engineering and Logistics Management at the Hong Kong University of Science and Technology. He received both his M.S. and Ph.D. in industrial and operations engineering from the University of Michigan, Ann Arbor. He is an associate editor of Technometrics, a department editor of the IIE Transactions, and on the editorial boards for the International Journal of Reliability, Quality and Safety Engineering (IJRQSE) and the International Journal of Six Sigma and Competitive Advantage (IJSSCA). He is an ASQ Certified Six Sigma Black Belt, ASQ authorized Six Sigma Master Black Belt Trainer, and former chair of the Quality, Statistics, and Reliability (QSR) Section at INFORMS. He is also the winner of the Best Paper Award for the IIE Transactions focus issue on Quality and Reliability in 2003. His research interests include quality engineering and management, statistical process control, monitoring and diagnosis.

Lesley Walls

University of Strathclyde
Department of Management Science
Glasgow, Scotland
lesley.walls@strath.ac.uk
http://www.managementscience.org/staff/lesley.asp

Chapter A.6

Lesley Walls has a Ph.D. (applied statistics). She is an IEC/TC56/WG2 expert and editorial board member of several reliability journals. Her research includes reliability modelling, business processes and risk assessment. She is a fellow of the UK Safety and Reliability Society, chartered statistician and was awarded the 2002 Simms prize by the Royal Aeronautical Society for REMM modelling research.

Wei Wang

Dana-Farber Cancer Institute
Department of Biostatistics and Computational Biology
Boston, MA, USA
wwang@jimmy.harvard.edu

Chapter C.21

Dr. Wang is assistant professor of biostatistics at Harvard School of Public Health and Dana-Farber Cancer Institute. She obtained her Ph.D. degree in statistics from the University of California at Davis. Dr. Wang's current research interests are mainly in developing semi-parametric and non-parametric methods in areas of survival analysis, longitudinal data analysis and functional data analysis. Dr. Wang also works at the statistical center of the Eastern Cooperative Oncology Group (ECOG) on collaborative research in cancer clinical trials.

Kenneth Williams

Yale University
Molecular Biophysics and Biochemistry
New Haven, CT, USA
kenneth.williams@yale.edu
http://info.med.yale.edu/wmkeck/

Chapter D.34

Dr. Williams received the Ph.D. degree in biochemistry from the University of Vermont in 1976. In 1980 he founded the Keck Laboratory (http://info.med.yale.edu/wmkeck/) and in 1986 he was one of the six founding members of the Association of Biomolecular ResourceFacilities (http://www.abrf.org/). He has 155 publications and directs the Yale/NHLBI Proteomics Center, NIDA Neuroproteomics Center, and the Proteomics Core of the Northeast Biodefense Center.

Richard J. Wilson

The University of Queensland
Department of Mathematics
Brisbane, Australia
rjw@maths.uq.edu.au
http://www.maths.uq.edu.au/~rjw

Chapter A.5

Dr. Wilson is a senior lecturer in statistics at The University of Queensland. His main research interests are in random processes, extremes and reliability, from both theoretical and applied statistics perspectives. Accordingly, he has worked on such diverse topics as modelling mineral phases in ores at the micro level, investigating warranty policies in manufacturing, exploring the relationship between the location of nerves to wisdom teeth and various factors, modelling wind downbursts, fitting models to significant wave height data and investigating aspects of the combustion of metal rods.

Baolin Wu

University of Minnesota, School of Public Health
Division of Biostatistics
Minneapolis, MN, USA
baolin@biostat.umn.edu
http://www.biostat.umn.edu/~baolin

Chapter D.34

Baolin Wu received the B.Sc. degree in probability and statistics from Beijng University in 1999 and the Ph.D. degree in biostatistics from Yale University in 2004. In 2004 he joined the Division of Biostatistics at the University of Minnesota as an assistant professor. His current research areas focus on computational biology and statistical learning.

Min Xie

National University of Singapore
Dept. of Industrial & Systems Engineering
Singapore, Singapore
mxie@nus.edu.sg
http://www.ise.nus.edu.sg/staff/xiemin/

Chapters A.3, B.16

Dr. Min Xie is a professor at National University of Singapore. He received his Ph.D. from Linköping University, Sweden in 1987 and has published over 100 articles in refereed journals and six books. He is an editor of International Journal of Reliability, Quality and Safety Engineering, a regional editor of Economic Quality Control, a department editor of IIE Transactions and associate editor IEEE Trans on Reliability. He is a fellow of IEEE.

Chengjie Xiong

Washington University in St. Louis
Division of Biostatistics
St. Louis, MO, USA
chengjie@wubios.wustl.edu
http://www.biostat.wustl.edu/faculty_staff/xiongc.shtml

Chapter C.19

Dr. Chengjie Xiong is a research assistant professor of biostatistics at Washington University School of Medicine. He received a B.S. in Mathematics from Xiangtan University (P.R. China), an M.S. in Applied Mathematics from Peking University (P. R. China), and a Ph.D. in statistics from Kansas State University in 1997. Dr. Xiong's research interests include statistical design of experiments, linear and nonlinear mixed models, longitudinal data analysis, survival analysis and reliability, categorical data analysis, order restricted statistical inferences, and their applications in medicine, biology, education, and engineering. Dr. Xiong has provided statistical consulting for researchers across the US in the areas of biology, medicine, agriculture, marketing and education and is the principal investigator of a NIH-funded project to study the statistical application in medical research. He is a member of the American Statistical Society.

Di Xu

Amercian Express
Dept. of Risk Management and Decision Science
New York, NY, USA
di.w.xu@aexp.com

Chapter B.12

Di Xu is a director in risk management and decision science in the consumer card services group at American Express. His research interests are multivariate statistical modelling, data mining, mathematical optimization, and their application in process control, product design, risk management and direct marketing acquisition. He graduated from Rutgers Univerity with a Ph.D. in Industrial Engineering in 2001.

Shigeru Yamada

Tottori University
Department of Social Systems Engineering
Tottori-shi, Japan
yamada@sse.tottori-u.ac.jp
http://www.sse.tottori-u.ac.jp/jouhou_source/hpsubmit/index.html

Chapter C.26

Dr. Yamada has been working as a professor in the Department of Social Systems Engineering at Tottori University, Japan, since 1993. He received his Ph.D. degree from Hiroshima University, Japan, in 1985. He has published numerous technical papers and books in the area of software reliability engineering, reliability engineering, and statistical quality. Dr. Yamada received the Best Author Award (1992) from the Information Processing Society of Japan, the TELECOM System Technology Award (1993) from the Telecommunications Advancement Foundation, the Best Paper Award (1999) from the Reliability Engineering Association of Japan, and the International Leadership Award in Reliability Engineering Research (2003) from the ICQRIT/SREQOM.

Jun Yan

University of Iowa
Department of Statistics and Actuarial Science
Iowa City, IA, USA
jyan@stat.uiowa.edu
http://www.stat.uiowa.edu/~jyan/

Chapter F.51

Dr. Jun Yan earned a Ph.D. in statistics from the Universityof Wisconsin – Madison in 2003. His research interests are functionaldata analysis, survival analysis, spatial statistics, statisticalcomputing, and cross-disciplinary statistical applications.

Shang-Kuo Yang

Department of Mechanical Engineering
National ChinYi Institute of Technology
Taiping City, Taiwan, R.O.C.
skyang@ncit.edu.tw
http://irw.ncit.edu.tw/mechanical/skyang/skyang.htm

Chapter E.44

Professor Yang received his B.S. in 1982 and the M.S. in 1985 in automatic control engineering from Feng Chia University, Taiwan. From 1985 to 1991 he was an assistant researcher and instrumentation system engineer of Flight Test Group, Aeronautic Research Laboratory, Chung Shan Institute of Science and Technology, Taiwan. Since 1991, he has been with the Department of Mechanical Engineering at National Chin Yi Institute of Technology, Taiwan, where he is a full professor and the chairperson. He received a Ph.D. in 1999 in mechanical engineering from National Chiao Tung University, Taiwan. His research interests are in reliability, data acquisition, and automatic control.

Kai Yu

Washington University in St. Louis, School of Medicine
Division of Biostatistics
St. Louis, MO, USA
yuka@mail.nih.gov

Chapter C.19

Dr. Yu is a research assistant professor at the Division of Biostatistics at Washington University, St. Louis. He obtained his Ph.D. in biostatistics from University of Pittsburgh in 2000. He completed a one-year postdoctoral training in statistical genetics in 2001 at Stanford University. His current research interests include biostatistics and genetic epidemiology.

Weichuan Yu

Yale Center for Statistical Genomics
and Proteomics, Yale University
Department of Molecular Biophysics
and Biochemistry
New Haven, CT, USA
weichuan.yu@yale.edu
http://noodle.med.yale.edu/~weichuan

Chapter D.34

Weichuan Yu received his Ph.D. degree in computer vision and image analysis from the University of Kiel, Germany in 2001. He was a postdoctoral associate at Yale University from 2001 to 2004. Currently he is a research faculty member in the Center for Statistical Genomics and Proteomics at Yale University. He is interested in computational analysis problems with biological and medical applications.

Panlop Zeephongsekul

Royal Melbourne Institute of Technology
University
School of Mathematical and Geospatial
Sciences
Melbourne, Victoria, Australia
panlopz@rmit.edu.au

Chapter F.52

Dr. Zeephongsekul received his B.Sc. degree with honors from Melbourne University and a Ph.D. degree in statistics from the University of Western Australia. He is currently an associate professor in the School of Mathematical and Geospatial Sciences at RMIT University, Melbourne, Australia. His research interests are broad, being in stochastic point processes, fuzzy sets, game theory, queuing theory and software reliability analysis. He has published in all those areas and his papers have appeared in many well-known international journals. He is also involved in many consulting projects with diverse clients, especially in applied statistics and the design and analysis of experiments.

Cun-Hui Zhang

Rutgers University
Department of Statistics
Piscataway, NJ, USA
czhang@stat.rutgers.edu
http://www.stat.rutgers.edu/people/faculty/zhang.html

Chapter E.40

Cun-Hui Zhang received his Ph.D. in statistics from Columbia University in 1984. He is currently a professor in the Department of Statistics at Rutgers University. His research interests include empirical Bayes, nonparametric and semiparametric methods, functional MRI, biased and incomplete data, networks, multivariate data, biometrics, and probability theory.

Heping Zhang

Yale University School of Medicine
Department of Epidemiology
and Public Health
New Haven, CT, USA
heping.zhang@yale.edu
http://peace.med.yale.edu

Chapter D.33

Heping Zhang is professor of biostatistics, child study, and statistics. He is interested in development of statistical methods and software and their applications in biomedical studies, particularly in behavioural science, epidemiology, genetics, psychiatry, and pregnancy outcomes. He publishes extensively on tree- and spline-based methods as well as latent variable models for genetic studies of ordinal traits. He is a fellow of the American Statistical Association.

Hongyu Zhao

Yale University School of Medicine
Department of Epidemiology
and Public Health
New Haven, CT, USA
hongyu.zhao@yale.edu
http://publichealth.yale.edu/faculty/zhao.html

Chapter D.34

Hongyu Zhao received a Ph.D. degree from the University of California at Berkeley in 1995. He is the Ira V. Hiscock Associate Professor at Yale University. He is interested in addressing statistical and computational problems in molecular biology and genetics. He has published more than 120 articles and is an associate editor for multiple journals including Biometrics and Statistica Sinica.

Kejun Zhu

China University of Geosciences
School of Management
Wuhan, Peoples Republic of China
zhubingl@pubilc.wh.hb.cn

Chapter C.19

Dr. Zhu's main area of research is soft computing where he has been working in the fields of system engineering and information systems. He is an associate editor of the Forecasting Journal. His current research activities include fuzzy systems, neural networks and genetic algorithms. He has presided over two programs of the National Natural Science Foundation of China.

Detailed Contents

Detailed Cont.

Part B Process Monitoring and Improvement

Part C Reliability Models and Survival Analysis

Part D Regression Methods and Data Mining

Subject Index

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z