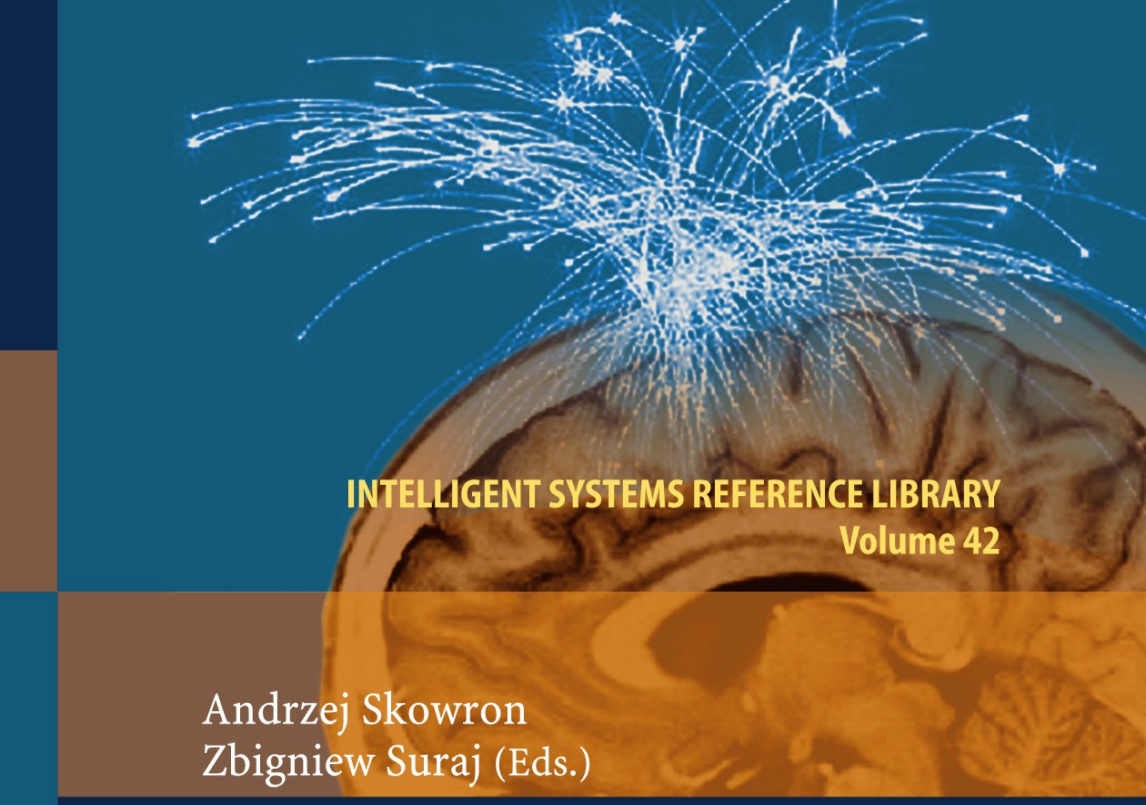


INTELLIGENT SYSTEMS REFERENCE LIBRARY
Volume 42

Andrzej Skowron
Zbigniew Suraj (Eds.)

Rough Sets and Intelligent Systems – Professor Zdzisław Pawlak in Memoriam

Volume 1

Springer

Intelligent Systems Reference Library 42

Editors-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Prof. Lakhmi C. Jain
School of Electrical and Information Engineering
University of South Australia
Adelaide
South Australia SA 5095
Australia
E-mail: Lakhmi.jain@unisa.edu.au

For further volumes:
http://www.springer.com/series/8578

Andrzej Skowron and Zbigniew Suraj (Eds.)

Rough Sets and Intelligent Systems – Professor Zdzisław Pawlak in Memoriam

Volume 1

Springer

Editors
Prof. Andrzej Skowron
Institute of Mathematics
University of Warsaw
Warsaw
Poland

Prof. Zbigniew Suraj
Institute of Computer Science
University of Rzeszów
Rzeszów
Poland

ISSN 1868-4394
ISBN 978-3-642-30343-2
DOI 10.1007/978-3-642-30344-9
Springer Heidelberg New York Dordrecht London

e-ISSN 1868-4408
e-ISBN 978-3-642-30344-9

Library of Congress Control Number: 2012939519

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Affiliations of Authors (Vol. 1 & Vol. 2)

Abdulaziz Alkhalid
Mathematical and Computer Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal, 23955-6900
Saudi Arabia
e-mail: abdulaziz.alkhalid@kaust.edu.sa

Talha Amin
Mathematical and Computer Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal, 23955-6900
Saudi Arabia
e-mail: talha.amin@kaust.edu.sa

Piotr Artiemjew
Department of Mathematics and Computer Science
University of Warmia and Mazury
54, Słoneczna, Olsztyn, 10-710
Poland
e-mail: artem@matman.uwm.edu.pl

S. Asharaf
Indian Institute of Management Kozhikode
Kozhikode, Kerala
India 673570
e-mail: asharaf@iimk.ac.in

Nouman Azam
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
e-mail: azam200n@cs.uregina.ca

Sanghamitra Bandyopadhyay
Machine Intelligence Unit
Indian Statistical Institute
203, B.T. Road, Kolkata 700108
India
e-mail: sanghami@isical.ac.in

Mohua Banerjee
Indian Institute of Technology
Kanpur, 208016
India
e-mail: mohua@iitk.ac.in

Minakshi Banerjee
Center for Soft Computing Research
Indian Statistical Institute
203, B.T. Road, Kolkata 700108
India
e-mail: mbanerjee23@gmail.com

Jan G. Bazan
Institute of Computer Science
University of Rzeszów
2, Dekerta, Rzeszów, 35-030
Poland
e-mail: bazan@univ.rzeszow.pl
and
Institute of Mathematics
University of Warsaw
2, Banacha., Warsaw, 02-097
Poland

Stanisława Bazan-Socha
Second Department of Internal Medicine
Jagiellonian University Medical College
8, Skawińska, Cracow, 31-066
Poland
e-mail: mmsocha@cyf-kr.edu.pl

Theresa Beaubouef
Department of Computer Science and Industrial Technology
SLU 10847
Southeastern Louisiana University
Hammond, LA 70402
USA
e-mail: tbeaubouef@selu.edu

Parag Bhalchandra
School of Computational Sciences
Swami Ramanand Teerth Marathwada University
Nanded, Maharashtra
India 431606
e-mail: srtmun.parag@gmail.com

Jerzy Błaszczyński
Institute of Computing Science
Poznań University of Technology
2, Piotrowo, Poznań, 60-965
Poland
e-mail: jblaszczynski@cs.put.poznan.pl

Zbigniew Bonikowski
Institute of Mathematics and Informatics
Opole University
Poland
e-mail: zbonik@math.uni.opole.pl

Edward Bryniarski
Institute of Mathematics and Informatics
Opole University
Poland
e-mail: edlog@math.uni.opole.pl

Sylwia Buregwa-Czuma
Institute of Computer Science
University of Rzeszów
2, Dekerta, Rzeszów, 35-030
Poland
e-mail: sczuma@univ.rzeszow.pl

Cory Butz
Department of Computer Science
University of Regina
3737 Wascana Parkway
Regina, Saskatchewan
Canada S4S 0A2
e-mail: cory.butz@uregina.ca

Mihir Kr. Chakraborty
Center for Cognitive Science
Jadavpur University
Kolkata 700032
India

and
Indian Statistical Institute
203, B.T. Road, Kolkata 700108
India
e-mail: mihirc4@gmail.com

Nick Cercone
Department of Computer Science and Engineering
York University Toronto
ON M3J 1P3, Canada
e-mail: ncercone@yorku.ca

Chien-Chung Chan
Department of Computer Science, CAS 221
University of Akron
Akron, OH 44325-4003
USA
e-mail: chan@uakron.edu

Chiao-Chen Chang
Department of International Business
National Dong Hwa University
No.1, Sec.2, Da Hsueh Rd., Shoufeng, Hualien 97401
Taiwan
e-mail: aka@mail.ndhu.edu.tw

Hongmei Chen
School of Information Science and Technology
Southwest Jiaotong University & Key Lab of Cloud Computing and Intelligent Technology
Chengdu, 610031
China
e-mail: hmchen@swjtu.edu.cn

Igor Chikalov
Mathematical and Computer Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal, 23955-6900
Saudi Arabia
e-mail: igor.chikalov@kaust.edu.sa

Yang-Chieh Chin
Department of Management Science
National Chiao Tung University
1001 Ta-Hsueh Road, Hsinchu 300
Taiwan
e-mail: jerry110888@gmail.com

Davydov Maksym Volodymyrovych
Lviv Polytechnic National University
12, Bandery, L'viv, 79013
Ukraine
e-mail: maks.davydov@gmail.com

Paweł Delimata
Institute of Computer Science
University of Rzeszów
2, Dekerta, Rzeszów, 35-030
Poland
e-mail: pdelimata@wp.pl

Ivo Düntsch
Brock University, St. Catharines, Ontario
Canada, L2S 3A1
e-mail: duentsch@brocku.ca

Aly A. Fahmy
Faculty of Computers and Information
Cairo University
Egypt
e-mail: aly.fahmy@gmail.com

Günther Gediga
Department of Psychology, Institut IV, Universität Münster
Fliednerstr. 21, Münster 48149
Germany
e-mail: gediga@uni-muenster.de

Salvatore Greco
Department of Economics and Business
University of Catania
55, Corso Italia, Catania, 95129
Italy
e-mail: salgreco@unict.it

Piotr Grochowalski
Institute of Computer Science
University of Rzeszów
2, Dekerta, Rzeszów, 35-030
Poland
e-mail: piotrg@univ.rzeszow.pl

Jerzy W. Grzymała-Busse
Department of Electrical Engineering and Computer Science
University of Kansas
3014 Eaton Hall
1520 W. 15th St., Lawrence, KS 66045-7621
USA
e-mail: jerzy@ku.edu
and
Institute of Computer Science, Polish Academy of Science
5, Jana Kazimierza, Warsaw, 01-237
Poland

Osman Gurdal
School of Medicine, IUPUI
975 W. Walnut Street, Indianapolis, IN 46202
USA
e-mail: ogurdal@iupui.edu

Aboul Ella Hassanien
Faculty of Computers and Information
Cairo University
Egypt
e-mail: aboitcairo@gmail.com

Hodych Oles Vasyliovych
Lviv Polytechnic National University
12, Bandery, L'viv, 79013
Ukraine
e-mail: oles.hodych@gmail.com

Jun Hu
Institute of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, 400065
P. R. China
e-mail: hujun@cqupt.edu.cn

Qinghua Hu
Harbin Institute of Technology
Box 458, Harbin 150001, Heilongjiang Province
P. R. China
e-mail: huqinghua@ hit.edu.cn

Shahid Hussain
Mathematical and Computer Sciences and Engineering Division
King Abdullah University of Science and Technology

Thuwal, 23955-6900
Saudi Arabia
e-mail: shahid.hussain@kaust.edu.sa

Masahiro Inuiguchi
Graduate School of Engineering Science
1-3 Machikaneyama-cho, Toyonaka, Osaka 560-8531
Japan
e-mail: inuiguti@sys.es.osaka-u.ac.jp

Ryszard Janicki
Department of Computing and Software
McMaster University
Hamilton, ON
Canada L8S 4K1
e-mail: janicki@mcmaster.ca

Janusz Kacprzyk
Systems Research Institute, Polish Academy of Sciences
6, Newelska, Warsaw, 01-447
Poland
e-mail: kacprzyk@ibspan.waw.pl
and
WIT - Warsaw School of Information Technology
6, Newelska, Warsaw, 01-447
Poland

Md. Aquil Khan
The Institute of Mathematical Sciences
Chennai, 600113
India
e-mail: mdaquilkhan@gmail.com
and
Institute of Mathematics
University of Warsaw
2, Banacha, Warsaw, 02-097
Poland

Ikno Kim
The Graduate School of Information, Production and Systems
Waseda University
2-7 Hibikino, Wakamatsu, Kitakyushu 808-0135
Japan
e-mail: octoberkim@hotmail.com

Beata Konikowska
Institute of Computer Science, Polish Academy of Sciences
5, Jana Kazimierza, Warszawa, 01-248
Poland
e-mail: beata.konikowska@ipipan.waw.pl

Maciej Kopczyński
Department of Computer Science
Białystok University of Technology
45A, Wiejska, Białystok, 15-351
Poland
e-mail: m.kopczynski@pb.edu.pl

Bożena Kostek
Multimedia Systems Department, Gdansk University of Technology
11/12, Narutowicza, Gdańsk, 80-233
Poland
e-mail: bozenka@sound.eti.pg.gda.pl

Marzena Kryszkiewicz
Institute of Computer Science
Warsaw University of Technology
15/19, Nowowiejska, Warsaw, 00-665
Poland
e-mail: mkr@ii.pw.edu.pl

Elżbieta Kubera
University of Life Sciences in Lublin
13, Akademicka, Lublin, 20-950
Poland
e-mail: elzbieta.kubera@up.lublin.pl

Leijun Li
Harbin Institute of Technology
Box 458, Harbin 150001, Heilongjiang Province
P. R. China
e-mail: lileijun1985@163.com

Tianrui Li
School of Information Science and Technology
Southwest Jiaotong University
Chengdu, 610031
China
e-mail: trli@swjtu.edu.cn

Chiun-Sin Lin
Department of Management Science
National Chiao Tung University
1001 Ta-Hsueh Road, Hsinchu 300
Taiwan
e-mail: netec@yahoo.com.tw

Tsau Young Lin
Department of Computer Science
San Jose State University
San Jose, CA 95192-0249
USA
e-mail: ty.lin@sjsu.edu

Pawan Lingras
Professor, Mathematics and Computing Science
Saint Mary's University, Halifax, Nova Scotia
Canada B3H3C3
e-mail: pawan.lingras@gmail.com

Kathy J. Liszka
Department of Computer Science, CAS 221
University of Akron
Akron, OH 44325-4003
USA
e-mail: liszka@uakron.edu

Mei-Chen Lo
Department of Business Management
National United University
No.1, Lienda Rd., Miaoli 36003
Taiwan
e-mail: meichen_lo@yahoo.com

Ilias Maglogiannis
University of Central Greece
Department of Computer Science and Biomedical Informatics
Papasiopoulou 2-4, PC 35100 Lamia
Greece
e-mail: imaglo@ucg.gr

Pradipta Maji
Machine Intelligence Unit
Indian Statistical Institute

203, B.T. Road, Kolkata 700108
India
e-mail: pmaji@isical.ac.in

A. Mani
Department of Pure Mathematics
University of Calcutta
35, Ballygunge Circular Road, Kolkata (Calcutta) 700019
India
e-mail: a.mani.cms@gmail.com

Victor W. Marek
Davis Marksbury Building
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0633
USA
e-mail: marek@cs.uky.edu

Shantan R. Marepally
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045
USA
e-mail: shantan@ku.edu

Barbara Marszał-Paszek
Institute of Computer Science, University of Silesia
39, Będzińska 39, Sosnowiec, 41–200
Poland
e-mail: bpaszek@us.edu.pl

Benedetto Matarazzo
Department of Economics and Business
University of Catania
55, Corso Italia, Catania, 95129
Italy
e-mail:matarazz@unict.it

Mikhail Moshkov
Mathematical and Computer Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal, 23955-6900
Saudi Arabia
e-mail: mikhail.moshkov@kaust.edu.sa

Kanlaya Naruedomkul
Department of Mahtematics, Faculty of Science
Mahidol University
Rama 6 Road, Rajadhevee, Bangkok, 10400
Thailand
e-mail: scknr@mahidol.ac.th

Hung Son Nguyen
Institute of Mathematics
University of Warsaw
2, Banacha, Warsaw, 02-097
Poland
e-mail: son@mimuw.ed.pl

Linh Anh Nguyen Institute of Informatics
University of Warsaw
2, Banacha Warsaw, 02-097
Poland
e-mail: nguyen@mimuw.edu.pl

Nikolski Iouri Volodymyrovych
Lviv Polytechnic National University
12, Bandery, L'viv, 79013
Ukraine
e-mail: y_nikol@yahoo.com

Hannu Nurmi
Public Choice Research Centre
Department of Political Science and Contemporary History
University of Turku
FIN-20014 Turku
Publicum (Assistentinkatu 7)
Finland
e-mail: hnurmi@utu.fi

Ewa Orłowska
National Institute of Telecommunications
1, Szachowa, Warsaw, 04-894
Poland
e-mail: E.Orlowska@itl.waw.pl

Sankar K. Pal
Center for Soft Computing Research
Indian Statistical Institute

203, B.T. Road, Kolkata 700108
India
e-mail: sankarpal@yahoo.com

Krzysztof Pancerz
Institute of Biomedical Informatics
University of Information Technology and Management
2, Sucharskiego, Rzeszów, 35-225
Poland
e-mail: kpancerz@wsiz.rzeszow.pl

Przemyslaw Wiktor Pardel
Institute of Computer Science
University of Rzeszów
2, Dekerta, Rzeszów, 35-030
Poland
e-mail: ppardel@univ.rzeszow.pl
and
Institute of Mathematics
University of Warsaw
2, Banacha, Warsaw, 02-097
Poland

Pasichnyk Volodymyr Volodymyrovych
Lviv Polytechnic National University
12, Bandery, L'viv, 79013
Ukraine
e-mail: vpasichnyk@gmail.com

Frederick E. Petry
Naval Research Lab
Geospatial Science and Technology Branch
Code 7444
Stennis Space Center, MS 39529
USA
e-mail: fpetry@nrlssc.navy.mil

Piotr Paszek
Institute of Computer Science, University of Silesia
39, Będzińska 39, Sosnowiec, 41–200
Poland
e-mail: paszek@us.edu.pl

Sushmita Paul
Machine Intelligence Unit
Indian Statistical Institute
203, B.T. Road, Kolkata 700108
India
e-mail: sushmita_t@isical.ac.in

Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta
Edmonton
Canada T6R 2G7
e-mail: wpedrycz@ualberta.ca

James F. Peters
Computational Intelligence Laboratory
Department of Electrical & Computer Engineering
University of Manitoba
75A Chancellor's Circle, EITC Room E1-526
Winnipeg, Manitoba
Canada R3T 3E2
e-mail: jfpeters@ee.umanitoba.ca

Lech Polkowski
Polish–Japanese Institute of Information Technology
86, Koszykowa, Warszawa, 02-008
Poland
e-mail: polkow@pjwstk.edu.pl

Astrid A. Prinz
Department of Biology
Emory University
O. Wayne Rollins Research Center
Room 2105, 1510 Clifton Road, Atlanta, GA 30322
USA
e-mail: astrid.prinz@emory.edu

Anna Maria Radzikowska
Faculty of Mathematics and Information Science
Warsaw University of Technology
1, Plac Politechniki, Warsaw, 00-661
Poland
e-mail: A.Radzikowska@mini.pw.edu.pl

Elisabeth Rakus-Andersson
Blekinge Institute of Technology
School of Engineering
S-37179 Karlskrona
Sweden
e-mail: Elisabeth.Andersson@bth.se

Zbigniew W. Raś
University of North Carolina
Dept. of Computer Science, Charlotte, NC 28223
USA
e-mail: ras@uncc.edu
and
Institute of Computer Science
Warsaw University of Technology
15/19, Nowowiejska, Warsaw, 00-665
Poland
and
Institute of Computer Science
Polish Academy of Sciences
5, Jana Kazimierza, Warsaw, 01-248
Poland

Grzegorz Rozenberg
Leiden Center for Natural Computing
Leiden University
Niels Bohrweg 1,
2333 CA Leiden
The Netherlands
e-mail: rozenber@liacs.nl

Mostafa A. Salama
Department of Computer Science
British University in Egypt
Cairo
Egypt
e-mail: mostafa.salama@gmail.com

Maria Semeniuk-Polkowska
Chair of Formal Linguistics
University of Warsaw
8–12, Browarna, Warszawa, 00-950
Poland
e-mail: m.polkowska@uw.edu.pl

Shcherbyna Yuri Mykolaiovych
Ivan Franko National University of L'viv
1, Universytetska, L'viv, 79000
Ukraine
e-mail: yshcherbyna@yahoo.com

Andrzej Skowron
Institute of Mathematics
University of Warsaw
2, Banacha, Warsaw, 02-097
Poland
e-mail: skowron@mimuw.ed.pl

Dominik Ślęzak
Institute of Mathematics
University of Warsaw
2, Banacha, Warsaw, 02-097
Poland
and
Infobright Inc.
34, lok. 219, Krzywickiego, Warsaw, 02-078
Poland
e-mail: slezak@{mimuw.edu.pl,infobright.com}

Roman Słowiński
Institute of Computing Science
Poznań University of Technology
2, Piotrowo, Poznań, 60-965
Poland
e-mail: rslowinski@cs.put.poznan.pl
and
Systems Research Institute, Polish Academy of Sciences
6, Newelska, Warsaw, 01-447
Poland

Tomasz G. Smolinski
Department of Computer and Information Sciences
Delaware State University
Science Center North
Room 344, 1200 N. DuPont Highway, Dover, DE 19901
USA
e-mail: tsmolinski@desu.edu

Barbara Sokołowska
Second Department of Internal Medicine
Jagiellonian University Medical College
8, Skawińska, Cracow, 31-066
Poland
e-mail: basiasok1@gmail.com

Omar S. Soliman
Faculty of Computers and Information
Cairo University
Egypt
e-mail: dr.omar.soliman@gmail.com

Sumalee Sonamthiang
Institute for Innovative Learning
Mahidol University
999 Phuttamonthon 4 Road, Nakhon Pathom, 73170
Thailand
e-mail:tew197@gmail.com

Jarosław Stepaniuk
Department of Computer Science
Białystok University of Technology
45A, Wiejska, Białystok, 15-351
Poland
e-mail: j.stepaniuk@pb.edu.pl

Zbigniew Suraj
Institute of Computer Science
University of Rzeszów
2, Dekerta, Rzeszów, 35-030
Poland
e-mail: zsuraj@univ.rzeszow.pl

Yu Ru Syau
Department of Information Management
National Formosa University
Huwei 63201, Yunlin
Taiwan
e-mail: yrsyau@nfu.edu.tw

Andrzej Szałas
Institute of Informatics
University of Warsaw
2, Banacha Warsaw, 02-097
Poland
e-mail: andrzej.szalas@mimuw.edu.pl
and
Dept. of Computer and Information Science
Linköping University
Linköping, SE-581 83
Sweden

Marcin Szeląg
Institute of Computing Science
Poznań University of Technology
2, Piotrowo, Poznań, 60-965
Poland
e-mail: mszelag@cs.put.poznan.pl

Tomasz Szmuc
Department of Applied Computer Science
AGH University of Science and Technology
30, Al. Mickiewicza, Cracow, 30-059
Poland
e-mail: tsz@agh.edu.pl

Marcin Szpyrka
Department of Applied Computer Science
AGH University of Science and Technology
30, Al. Mickiewicza, Cracow, 30-059
Poland
e-mail: mszpyrka@agh.edu.pl

Hongmei Tian
School of Information Science and Technology
Southwest Jiaotong University
Chengdu, 610031
China
e-mail: 470777848@qq.com

Li-Shiang Tsay
School of Technology
213 Price Hall, NC A&T State University, Greensboro NC 27411
USA
e-mail: ltsay@ncat.edu

Shusaku Tsumoto
Department of Medical Informatics
Shimane University
School of Medicin
89-1 Enya-cho, Izumo-city, Shimane 693-8501
Japan
e-mail: tsumoto@med.shimane-u.ac.jp

Gwo-Hshiung Tzeng
Department of Business Administration
Kainan University
No. 1 Kainan Road, Luchu, Taoyuan 338
Taiwan
e-mail: ghtzeng@mail.knu.edu.tw
and
Institute of Management of Technology
National Chiao Tung University
1001 Ta-Hsueh Road, Hsinchu 300
Taiwan
e-mail: ghtzeng@cc.nctu.edu.tw

Alicja Wakulicz-Deja
Institute of Computer Science, University of Silesia
39, Będzińska 39, Sosnowiec, 41–200
Poland
e-mail: wakulicz@us.edu.pl

Guoyin Wang
Chongqing University of Posts and Telecommunications
Chongqing, 400065
P. R. China
e-mail: wanggy@ieee.org

Anita Wasilewska
Department of Computer Science
Stony Brook University
Stony Brook, NY
USA
e-mail: anita@cs.sunysb.edu

Piotr Wasilewski
Institute of Informatics
University of Warsaw
2, Banacha, Warsaw, 02-097
Poland
e-mail: piotr@mimuw.edu.pl

Junzo Watada
The Graduate School of Information, Production and Systems
Waseda University
2-7 Hibikino, Wakamatsuku, Kitakyushu 808-0135
Japan
e-mail: junzow@osb.att.ne.jp

Alicja A. Wieczorkowska
Polish-Japanese Institute of Information Technology
86, Koszykowa 86,Warsaw, 02-008
Poland
e-mail: alicja@poljap.edu.pl

Marcin Wolski
Department of Logic and Philosophy of Science
Maria Curie-Skłodowska University
4, Pl. Marii Curie-Skłodowskiej, Lublin, 20-031
Poland
e-mail: marcin.wolski@umcs.lublin.pl

Urszula Wybraniec-Skardowska
Group of Logic, Language and Information
Opole University
Poland
e-mail: uws@uni.opole.pl

JingTao Yao
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
e-mail: jtyao@cs.uregina.ca

Yiyu Yao
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
e-mail: yyao@cs.uregina.ca

Sławomir Zadrożny
Systems Research Institute, Polish Academy of Sciences
6, Newelska, Warsaw, 01-447
Poland
e-mail: zadrozny@ibspan.waw.pl

and
Technical University of Radom
29, Malczewskiego, Radom, 26-600
Poland

Pengfei Zhu
Harbin Institute of Technology
Box 458, Harbin 150001, Heilongjiang Province
P. R. China
e-mail: zhupengfeifly@gmail.com

Wojciech Ziarko
Department of Computer Science
University of Regina
Regina, Saskatchewan, S4S 0A2
CANADA
e-mail: ziarko@sasktel.net

Beata Zielosko
Mathematical and Computer Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal, 23955-6900
Saudi Arabia
e-mail: beata.zielosko@kaust.edu.sa
and
Institute of Computer Science
University of Silesia
39, Będzińska St., Sosnowiec, 41-200
Poland

To the memory of

Professor Zdzisław I. Pawlak

Foreword

This book is a tribute to the late Professor Zdzisław Pawlak (1926-2006), a founder of the Polish school of Artificial Intelligence, and one of pioneers of Computing Science in Poland and worldwide.

The work of Professor Pawlak fits perfectly with the definition of Computing Science given by Peter J. Denning (2000): "Computing Science is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application". In such an understanding of Computing Science, the contribution of Professor Pawlak was without a doubt most seminal among all Polish scientists.

This book is an edited collection of chapters written by authors who knew Professor Pawlak. The first chapter comprises an introduction into his life and achievements through testimonies of his students and collaborators. The remaining chapters refer to his research, and show its far reaching consequences for today's computing practices.

Professor Pawlak's scientific contributions were of a fundamental nature and established new directions in information processing research. His foremost contribution to the body of scientific knowledge was the formulation of Rough Set theory - a powerful yet simple and general concept for reasoning about data.

It was Paul Valéry who said, "A vague fact may be more perfidious than erroneous reasoning". Indeed, an error in reasoning can be found and corrected while a vague fact cannot be corrected and when put together with certain facts, it may undermine the entire reasoning process. Therefore, vague facts have to be identified as a part of knowledge about the given reality so they can be used in approximate reasoning. Rejection of these facts is equally wrong as is ignoring their vagueness, since vagueness may be symptomatic for very important aspects of the considered reality.

Rough Set theory perfectly addresses this issue by permitting to differentiate between vague and certain facts when reasoning about data. The theory is based on an observation that knowledge about the objects of a universe is granular - objects that are described with the same information are indiscernible and form granules of knowledge about the universe. It is, however, not always possible to precisely

express a concept concerning a given set of objects in terms of knowledge about the universe, using the available granules of knowledge. Then, such a concept needs to be expressed approximately by two sets of objects called lower and upper approximations, respectively. The lower approximation is a union of all the granules that are completely included in the concept, while the upper approximation is a union of all the granules that have a nonempty intersection with the concept. The difference between upper and lower approximations is called a boundary region and it consists of vague objects (facts) - those where it is impossible to state with certainty if they belong to the considered concept. This distinction between certain knowledge (lower approximation) and doubtful knowledge (boundary region) has a crucial impact on the reasoning process. Rough Set reasoning about data allows for the discovery of interesting relationships in data, the reduction of data, measuring the value of information carried by data, and the induction of decision rules that are helpful for explaining relationships in data to be used in decision support and in the construction of a strategy of intervention.

Rough set theory is complementary to Fuzzy Set theory and Possibility theory. Together, these three theories provide the best available methodologies for the analysis of data that bears some kind of "imperfection" such as vagueness, ambiguity, imprecision, incompleteness, and uncertainty.

The first paper on rough sets by Professor Pawlak, published in 1982 in the Journal of Information and Computer Science, was quoted 4150 times (per the SCOPUS database), and the number of papers that have the term "rough set" in their title, abstract or as a keyword is well over 10,000, with a Hirsch index greater than 85. These papers are about theoretical extensions to the rough set concept and also about applications in numerous areas. This sheer volume of publications is a powerful testimony to the fact that Rough Set theory is a very important contribution when answering questions in Artificial Intelligence, Decision theory, Operational Research, Theory of Conflicts, Control theory, Machine Learning and Knowledge Discovery, to name the few.

On a personal note, I am grateful to Professor Pawlak for teaching me Rough Set theory in the early stages of its conception. Together with my brother Krzysztof, a surgeon, we had the privilege of working with Professor Pawlak on the first practical application of the theory - verification of indications for the treatment of duodenal ulcer by HSV. My cooperation with Professor Pawlak lasted until his health permitted, and expanded to include my younger colleagues, like Salvatore Greco. Professor Pawlak's pleasant personality made this collaboration a true pleasure and also attracted other researchers to this field. I also fondly remember talking to Professor Pawlak about things other than research - especially about photography. He was an accomplished photographer and also a landscape painter with an acute ability to capture the interplay of light and shadow.

Despite his overwhelming scientific stature, Professor Zdzisław Pawlak was a modest and considerate person. He radiated enthusiasm and kind encouragement for the young. For all these reasons, we wish to keep the memory of him alive.

We are very grateful to the editors of this volume - Professors Andrzej Skowron and Zbigniew Suraj - for undertaking the task of preparing this special book. It will certainly contribute to the memory of Professor Pawlak as a person and as a scientist.

Let me finish this foreword with a quote from Hippocrates that very well describes Professor Pawlak's life and achievements - "Vita brevis, ars longa" (Life elapses, art lasts forever). Indeed, Professor Pawlak's accomplishments transcend his life.

Roman Słowiński
Poznań, March 2012

Preface

This book is dedicated to the memory of Professor Zdzisław Pawlak, who passed away almost six years ago. He is the founder of the Polish school of Artificial Intelligence and one of the pioneers in Computer Engineering and Computer Science with worldwide influence. He was a truly great scientist, researcher, teacher and a human being.

Professor Pawlak's most important discovery was his invention of the rough set theory in 1982, which gained vast popularity throughout the World. More than 5000 English-language publications and also more than 5000 Chinese-language publications about Pawlak's theory and its applications have been published so far, including many books. Those publications include both specializations and extensions of rough set theory. Their goal is to solve new scientific problems, examining connections between the theory and other approaches and dealing with applications of the theory in practice. Moreover, a number of books devoted to rough sets theory were published worldwide. Numerous conferences, *e.g.*, in China, India, Japan, Canada, USA, in Europe and also recently in Australia and Africa, were organized. The rough sets theory has an immense following in China and research on rough sets also is significantly growing in India.

Rough set theory has attracted worldwide attention with many researchers and practitioners, who have contributed essentially to its development and applications. Rough set theory overlaps with many other theories. Despite this, rough set theory may be considered as an independent discipline in its own right. The rough set approach seems to be of fundamental importance in artificial intelligence and cognitive sciences, especially in research areas such as adaptive and autonomous systems, bioinformatics, data mining and knowledge discovery, decision analysis, expert systems, machine learning, intelligent systems, inductive reasoning, pattern recognition, mereology, digital image processing, digital image analysis and signal analysis. A wide range of applications of methods based on rough set theory alone or in combination with other approaches have been discovered in many areas including: acoustics, bioinformatics, business and finance, chemistry, computer engineering (*e.g.*, data compression, digital image processing, digital signal processing, parallel and distributed computer systems, sensor fusion, fractal engineering),

decision analysis and systems, economics, electrical engineering (*e.g.*, control, signal analysis, power systems), environmental studies, digital image processing, informatics, medicine, molecular biology, musicology, neurology, robotics, social science, software engineering, spatial visualization, Web engineering, and Web mining.

Professor Pawlak inspired many computer scientists and mathematicians both in both Poland and throughout the world. His students and collaborators created research teams in many countries, including, besides of his native Poland, the United States, Canada, Japan, Norway, Sweden and other places. It would be hardly possible t o find a computer science institution, in his native Poland without encountering a faculty influenced by Professor Pawlak. His scientific achievements continue to inspire his many students still working in these institutions and also the next generations of students.

This book prepared in two volumes contains more than 50 chapters. This demnonstrates that the scientific approaches discovered by of Professor Zdzisław Pawlak, especially the rough set approach as a tool for dealing with imperfect knowledge, are vivid and intensively explored by many researchers in many places throughout the world. The submitted papers prove that interest in rough set research is growing and is possible to see many new excellent results both on theoretical foundations and applications of rough sets alone or in combination with other approaches.

The book is divided into two volumes.

The first volume contains the following chapters.

Chapter "Professor Zdzisław Pawlak (1926-2006): Founder of the Polish School of Artificial Intelligence" by Andrzej Skowron, Mihir Kr. Chakraborty, Jerzy Grzymła-Busse,Victor Marek, Sankar K. Pal, James Peters, Grzegorz Rozenberg, Dominik Ślęzak, Roman Słowiński, Shusaku Tsumoto, Alicja Wakulicz-Deja, Guoyin Wang, and Wojciech Ziarko is dedicated to the memory of Professor Zdzisław Pawlak, founder of the Polish school of Artificial Intelligence and one of the pioneers in Computer Engineering and Computer Science with worldwide influence. In particular, it contains a few selected speech fragments pointing to Professor's scientific achievements along with personal comments by Andrzej Skowron, one of the authors, on features of Professor Pawlak as a truly great scientist, teacher and a human being.

The list of works by Professor Zdzisław Pawlak, prepared by Andrzej Skowron, is included in the chapter "List of Works by Professor Zdzisław Pawlak (1926-2006)".

Chapter "Rough Sets: From Rudiments to Challenges" by Hung Son Nguyen and Andrzej Skowron contains a survey about rough sets together with comments on possible future research directions and challenges.

In Chapter "Zdzisław Pawlak, Databases and Rough Sets", Victor W. Marek presents work of Zdzisław Pawlak in the area of databases and the extension of that work to the theory of rough sets. In particular, the author concentrates on motivations of Professor Pawlak for introducing information storage and retrieval systems and describes how this, eventually, led to rough sets theory.

Chapter "jMAF - Dominance-based Rough Set Data Analysis Framework" by Jerzy Błaszczyński, Salvatore Greco, Benedetto Matarazzo, Roman Słowiński, and Marcin Szeląg, presents a rough set data analysis software jMAF. It employs java

Rough Set (jRS) library in which are implemented data analysis methods provided by the (variable consistency) Dominance-based Rough Set Approach (DRSA). The chapter also provides some basics of the DRSA and of its variable consistency extension.

Chapter "Dynamic Programming Approach for Exact Decision Rule Optimization" by Talha Amin, Igor Chikalov, Mikhail Moshkov, and Beata Zielosko, discusses an extension of dynamic programming approach to sequential optimization of exact decision rules relative to the length and coverage. The chapter also contains results of experiments with decision tables from UCI Machine Learning Repository.

Chapter "Approaches for Updating Approximations in Set-Valued Information Systems While Objects and Attributes Vary with Time " by Hongmei Chen, Tianrui Li, and Hongmei Tian focuses on studying principles for incrementally updating approximations in a set-valued information system while attributes and objects are added. Methods for updating approximations of concepts in set-valued information systems with attributes and objects changing simultaneously are presented. Experimental evaluation of the proposed methods is included.

Ivo Düntsch and Günther Gediga describe in Chapter "On the gradual evolvement of things" the generic properties of a visual system without hard coding the environment. As a measure of approximation, Pawlak's approximation quality is used. The considerations are related to some ideas of the Gibson ecological approach to perception.

The Chapter "On Empirical Comparison of Rule Sets Induced by LERS and Probabilistic Rough Classification" by Jerzy W. Grzymała-Busse, Shantan R. Marepally, and Yiyu Yao explores an extension of rough set theory, based on probability theory. In particular, parameterized approximations are used together with the corresponding positive, boundary, and possible rules. The results of parameter tuning on the quality of the induced classifiers based on such rules are reported.

In Chapter "Exploring Neighborhood Structures with Neighborhood Rough Sets in Classification Learning", Qinghua Hu, Leijun Li, and Pengfei Zhu introduce neighborhoods of samples to granulate the universe and use the neighborhood granules to approximate classification, thus a model of neighborhood rough sets is derived. Some machine based on the model learning algorithms, including boundary sample selection, feature selection and rule extraction, are developed.

Chapter "Rough Representations of Ill-Known Sets and Their Manipulations in Low Dimensional Space" by Masahiro Inuiguchi, focuses on investigations of the rough representations of graded ill-known sets and the manipulations of possibility and necessity measures of graded ill-known sets using general conjunction and implication functions in the universe.

In Chapter "Property-Driven Rough Sets Approximations of Relations" by Ryszard Janicki, a problem of approximating an arbitrary relation by a relation with desired properties is formally defined and analysed. The concepts of α-lower and upper approximatios are introduced and their properties are discussed. Two special cases, approximation by partial orders and approximation by equivalence relations are discussed in detail.

Chapter "Towards a Comprehensive Similarity Analysis of Voting Procedures Using Rough Sets and Similarity Measures" by Janusz Kacprzyk, Hannu Nurmi, and Sławomir Zadrożny, presents and approach to the evaluation of similarity of voting procedures with respect to a set of criteria which are widely used in the social choice literature. First, a qualitative rough sets based analysis is proposed, and then an additional quantitative analysis is added by using two measures of similarity of binary patters widely employed in many areas , i.e. the one based on the Hamming distance and the one due to Jaccard-Needham. The approach proposed constitutes a step towards the solutution of a difficult problem of determining the (degree of) similarity of voting procdures by providing a comprehensive qualitative and quantitative view.

Chapter "Algebras for Information Systems" by Md. Aquil Khan and Mohua Banerjee, introduces algebraic structures for different kinds of information systems together with the representation theorems for classes of algebras corresponding to these structures. Finally, equational logics for deterministic, incomplete and non-deterministic information systems are presented.

Chapter "DNA Rough-Set Computing in the Development of Decision Rule Reducts" by Ikno Kim, Junzo Watada, and Witold Pedrycz, introduces a DNA rough-set computation technique for dealing with the NP-hard problem problem of optimal reduction of decision rules. The proposed technique is a composition of computational DNA molecular techniques and is effectively employed to alleviate the computational complexity of the considered optimization problem.

Chapter "Three-valued Logic for Reasoning about Covering-Based Rough Sets" by Beata Konikowska, introduces a tool for reasoning about covering-based rough sets in the form of three-valued logic with logical values corresponding to positive, negative region and the boundary regions of a set. The author presents a strongly sound sequent calculus for this logic, together with the proof of strong completeness for a subset of its language.

In Chapter "Music Information Retrieval in Music Repositories" by Bożena Kostek, the key concepts associated with automated music information retrieval and music recommendation are discussed. Experiments on a constructed music database with different kinds of classifiers are reported. A proposal for music retrieval and annotation aided by gaze tracking is also discussed.

In Chapter "Rough Support Vectors: Classification, Regression, Clustering" by Pawan Lingras, Parag Bhalchandra, Cory Butz, and S. Asharaf, it is shown that the concepts of margins in support vector techniques provides a natural relationship with the rough set theory. The authors describe rough set theoretic extensions of support vector technologies for classification, prediction, and clustering. The theoretical formulations of rough support vector machines, rough support vector regression, and rough support vector clustering are supported with a summary of experimental results.

Chapter "Logic-based Roughification" by Linh Anh Nguyen and Andrzej Szałas includes novel roughification techniques for constructing equivalence/similarity

relations adequate for Pawlak-like approximations. The authors also present applications of the proposed approach in granulating relational databases as well as concept learning and approximation in description logic-based information systems.

Chapter "How Near Are Zdzisław Pawlak's Paintings? Merotopic Distance Between Regions-of-Interest" by James F. Peters, presents an approach to measuring the nearness of Pawlak's paintings in terms of the merotopic distance between collections of neighborhoods in image regions-of-interest.

Chapter "An Implementation of the Zdzisław Pawlak Idea for Reasoning about Uncertainty. Approximate Reasoning by Parts" by Lech Polkowski and Maria Semeniuk-Polkowska, presents a mereological calculus of parts, in which concepts become elementary objects and relations among them are expressed as relations of being parts to degrees. This analysis allows, in particular, for approximations to various degrees. A characterization of continuous rough inclusions parallel to the Menu-Pavelka characterization of continuous t–norms is proposed.

Chapter "Granular Concept Mapping and Applications" by Sumalee Sonamthiang, Kanlaya Naruedomkul, and Nick Cercone, presents a granular concept hierarchy (GCH) construction and mapping of the hierarchy for granular knowledge. A granule description language and granule measurements are proposed to enable mapping for an appropriate granular concept that represents sufficient knowledge to solve the problem at hand. Applications of GCH are demonstrated through learning of higher order decision rules.

Chapter "Rough Sets and Medical Differential Diagnosis" by Shusaku Tsumoto, discusses a correspondence between the core ideas of rough sets and medical differential diagnosis.

In Chapter "Science and Semantics: A Note on Rough Sets and Vagueness" by Marcin Wolski, rough set theory is presented against the background of recent philosophical discussions about vagueness and empirical sciences.

The second volume contains the following chapters.

In chapter "From Logic to Computer Science – A Personal Experience", Anita Wasilewska explains why she is so grateful to Professor Zdzisław Pawlak whom was possible to meet on her way from mathematics to computer science.

Chapter "Knowledge algebras and their discrete duality" by Ewa Orlowska and Anna Maria Radzikowska introduces a class of algebras referred to as knowledge algebras and a class of knowledge frames. Representation theorems for these classes leading to a discrete duality are proved.

Chapter "Comparison of Greedy Algorithms for Decision Tree Optimization" by Abdulaziz Alkhalid, Igor Chikalov, and Mikhail Moshkov is devoted to the comparison of 16 types of greedy algorithms for decision tree construction with optimal decision trees generated by the dynamic programming approach. Optimization is performed relative to minimal values of different parameters of decision trees. The results of experiments are reported and discussed.

In Chapter "A Review of the Knowledge Granulation Methods: Discrete vs Continuous Algorithms" by Piotr Artiemjew, rough inclusions and some of their weaker variants are used to define similarity relations. Applications to classification problems are discussed.

In Chapter "Game-theoretic Rough Sets for Feature Selection" by Nouman Azam and JingTao Yao, a game-theoretic rough sets based method is formulated for selecting important features by combining multiple measures representing importance levels for a feature. The method incorporates the measures as players in a game where each player employs a three-way decision in selecting features. The included demonstrative example suggests that this method may be useful for feature selection in text categorization.

Chapter "A Clustering Approach to Image Retrieval Using Range Based Query and Mahalanobis Distance" by Minakshi Banerjee, Sanghamitra Bandyopadhyay, and Sankar K. Pal, puts forward a new approach to address a general purpose content-based image retrieval task. The effectiveness of the proposed algorithm is demonstrated with increased accuracy and reduced retrieval time.

Chapter "Classifiers Based on Data Sets and Domain Knowledge: A Rough Set Approach" by Jan G. Bazan, Stanisława Bazan-Socha, Sylwia Buregwa-Czuma, Przemysław Pardel, Andrzej Skowron, and Barbara Sokolowska, presents the ontology approximation method for inducing complex classifiers from experimental data and domain knowledge. The experimental results on different data sets are reported, in particular on (i) data set generated by a vehicular traffic simulator, (ii) real-life data set concerning coronary heart disease obtained from Second Department of Internal Medicine, Jagiellonian University Medical College, Cracow, Poland, and (iii) real-life data set concerning respiratory failure obtained from Neonatal Intensive Care Unit in the Department of Pediatrics, Jagiellonian University Medical College, Cracow, Poland.

Chapter "Incorporating Rough Data in Database Design for Imprecise Information Representation" by Theresa Beaubouef and Frederick E. Petry, provides discussions of how it is possible to design relational databases to allow the incorporation of uncertain data characterized using rough set theory. This included Entity-Relationship modeling, rough functional dependencies and rough normal forms. Security issues as dealt with in statistical databases are also discussed as well as an example of the representation of uncertain spatial data by rough sets.

The approach based on a pragmatic view of representation of knowledge is used in Chapter 'Rough Pragmatic Description Logic" by Zbigniew Bonikowski, Edward Bryniarski, and Urszula Wybraniec-Skardowska for introducing and for investigation of a rough description logic.

Chapter "Application of Rough Set Theory to Sentiment Analysis of Microblog Data" by Chien-Chung Chan and Kathy J. Liszka, presents the use of rough set theory to formulate sentimental approximation spaces based on key words for assessing sentiment of microblogging messages. The sentimental approximation space provides contextual sentiment from the entire collection of messages, and it enables the evaluation of sentiment of different subjects, not in isolation, but in context. The sentimental approximation space offers potentially more insightful information about a subject than simple polarity answers of positive or negative.

Chapter "Relationships for Cost and Uncertainty of Decision Trees" by Igor Chikalov, Shahid Hussain, and Mikhail Moshkov, presents the results of studies on the relationships between the cost and the uncertainty of decision trees as well as on

the relationships between the number of nodes and the depth in the case of exact decision trees. The developed tools are based on dynamic programming approach and are applicable only to relatively small decision tables. The results of experiments are reported and discussed.

Chapter "The Impact Rules of Recommendation Sources for Adoption Intention of Micro-Blog Based on DRSA with Flow Network Graph" by Yang-Chieh Chin, Chiao-Chen Chang, Chiun-Sin Lin, and Gwo-Hshiung Tzeng, focuses on the micro-blog (*i.e.*, a new communication channel with which people share short text messages on public and private networks) on Facebook. The main purpose of the reported study is to explore and compare what recommendation sources influence the intention to use micro-blogs and to combine the personal characteristics/ attributes of gender, daily internet hour usage and past use experience to infer the usage of micro-blogs decision rules using a dominance-based rough-set approach (DRSA) with flow network graph.

Chapter "Providing Feedback in Ukrainian Sign Language Tutoring Software" by M.V. Davydov, I.V. Nikolski, V.V. Pasichnyk, O.V. Hodych, and Y.M. Scherbyna, focuses on video recognition methods implemented as part of the Ukrainian Sign Language Tutoring Software. The proposed software system for sign language recognition supports user interaction with the system during learning of signs and the verification process. The developed feedback mechanism significant improves the training experience for users. The results of experiments are reported and discussed.

Chapter "Hybrid Methods in Data Classification and Reduction" by Paweł Delimata and Zbigniew Suraj summarizes numerous results of the authors on issues of data reduction, feature subset selection and classifier construction. In particular, applications of reducts, deterministic and inhibitory decision rules for feature selection are presented.

Chapter "Uncertainty Problem Processing with Covering Generalized Rough sets" by Jun Hu and Guoyin Wang, focuses on applications of the covering generalized rough set approach. There are proposed two models: knowledge reduction model and covering generalized rough fuzzy model.

Chapter "Hardware Implementations of Rough Set Methods in Programmable Logic Devices" by Maciej Kopczynski and Jarosław Stepaniuk, discusses existing results on hardware realization of rough set algorithms in the Field Programmable Gate Array (FPGA) logic devices.

In Chapter "Determining Cosine Similarity Neighborhoods by Means of the Euclidean Distance", Marzena Kryszkiewicz presents a scalable method for computing cosine similarity neighborhoods of vectors by employing the Euclidean distance applied to $(\alpha-)$normalized forms of these vectors and the triangle inequality. There are considered three types of sets of cosine similar vectors: all vectors the similarity of which to a given vector is not less than an ε threshold value and two variants of k-nearest neighbors of a given vector.

Chapter "Time Variability-Based Hierarchic Recognition of Multiple Musical Instruments in Recordings" by Elżbieta Kubera, Alicja A. Wieczorkowska, and Zbigniew W. Raś, focuses on automatic identification of musical instruments in

polyphonic audio recordings. The reported experiments demonstrate that the performance of classifiers enhanced, in particular by new temporal parameters introduced to supply additional temporal information for the timbre recognition, lead to improvement of the classifier performance.

In chapter "Unifying Variable Precision and Classical Rough Sets: Granular Approach" by Tsau Young Lin and Yu Ru Syau it is shown that neighborhood systems (NS) can integrate Ziarko's variable precision rough set model (VPRSM) and Pawlak 's classical rough sets into one concept.

Chapter "Fuzzy Hybrid MCDM for Building Strategy Forces" by Mei-Chen Lo and Gwo-Hshiung Tzeng, adopts Fuzzy multiple criteria decision making methods and discuss how the technology, marketing and research & development forces operate and suggests ways of adjusting to them, and, where possible, of taking advantage of them.

Chapter "Rough Set Based Feature Selection: Criteria of Max-Dependency, Max-Relevance, and Max-Significance" by Pradipta Maji and Sushmita Paul, reports a rough set based feature selection algorithm called maximum relevance - maximum significance (MRMS), and its applications on quantitative structure activity relationship (QSAR) and gene expression data. The importance of rough set theory for computing both relevance and significance of the features is also established. The results of experiments are reported and discussed.

In Chapter "Towards Logics of Some Rough Perspectives of Knowledge" by A. Mani, semantic frameworks for dealing with such issues as concepts of relative consistency of knowledge, conflict representation and resolution are introduced and developed. The proposed semantics may be of interest for multi-agent systems, dynamic spaces and collections of general approximation spaces.

In Chapter "Classifiers Based on Nondeterministic Decision Rules" by Barbara Marszał-Paszek and Piotr Paszek, classifiers based on rough set theory and nondeterministic decision rules are discussed. The reported experiments are showing that enhancing rule-based classifiers with nondeterministic rules may lead to increasing of the classification quality.

In Chapter "Approximation and Rough Classification of Letter-like Polygon Shapes" by Elisabeth Rakus-Andersson, is presented a rough set based method to classification of discrete two-dimensional point sets resembling some letters together with a rough set technique for verifying decisions about the primary recognitions of the curves' appearance as letter shapes. The results are utilized in the classifications of internet packet streams or in the analysis of wave signals typical of, *e.g.*, medical examinations.

Chapter "Rough Set-based Identification of Heart Valve Diseases Using Heart Sounds" by Mostafa A. Salama, Omar S. Soliman, Ilias Maglogiannis, Aboul Ella Hassanien, and Aly A. Fahmy, presents an application of the rough set approach to classification of heart sound diseases using heart sounds. The reported experiments are showing that the rough set based approach outperforms several other well known machine learning techniques.

In Chapter "Rough Sets and Neuroscience" by Tomasz G. Smolinski and Astrid A. Prinz, examples of the existing and potential applications of rough set theory (and

its hybridizations) in neuroscience and neurology are presented. Moreover, a discussion of further development of rough-neural computing, stimulated by relationships of rough sets with neuroscience, is provided.

In Chapter "On Knowledge Representation and Automated Methods of Searching Information in Bibliographical Data Bases: A Rough Set Approach" by Zbigniew Suraj, Piotr Grochowalski, and Krzysztof Pancerz, is presented an approach to searching for information in bibliographical data bases founded on rough set theory and the domain knowledge represented by ontologies. The reported experiments are performed on data gathered in the Rough Set Database System (RSDS).

In Chapter "Design and Verification of Rule-Based Systems for Alvis Models" by Marcin Szpyrka and Tomasz Szmuc, is presented a method of encoding and verification of rule-based systems with the Haskell functional language in order to include them into Alvis, a modeling language designed for embedded systems that provides a possibility of a formal model verification.

Chapter "On Objective Measures of Actionability in Knowledge Discovery" by Li-Shiang Tsay and Osman Gurdal, is included a rough set method for generating a set of rules by utilizing the domain experts' prior knowledge to formulate its inputs and to evaluate the observed regularities it discovers. The generated rule overcomes the traditional data-centered pattern mining resulting to bridge the gap and enhance real-world problem solving capabilities.

In Chapter "Pseudometric Spaces from Rough Sets Perspective" by Piotr Wasilewski, relationships between approximation spaces and pseudometric spaces are presented. Investigations are focused on the class of pseudometric spaces which are lower bounded in each point since open sets in these spaces coincide with definable sets of some prescribed approximation spaces.

Editors of this book are proud to present the readers this book.

Andrzej Skowron and Zbigniew Suraj
Warszawa, Rzeszów, March 2012

Acknowledgement

We are greatly indebted to all Authors of chapters submitted to this book, the Members of Editorial Committee, i.e., Janusz Kacprzyk, Sankar K. Pal, Roman Słowiński, Dominik Ślęzak, Shusaku Tsumoto, Guoyin Wang, Wojciech Ziarko who kindly joint us in the work on the book and to all reviewers of submitted papers including Mohua Banerjee, Jan Bazan, Anna Gomolińska, Salvatore Greco, Piotr Hońko, Andrzej Janusz, Beata Konikowska. Victor Marek, Hung Son Nguyen, Tuan Trung Nguyen, Witold Pedrycz, James Peters, Lech Polkowski, Sheela Ramanna, Piotr Synak, Dominik Ślęzak, Piotr Wasilewski, Alicja Wieczorkowska, Marcin Wolski for their work and help in making this important book available.

We extend an expression of gratitude to Professors Janusz Kacprzyk and Lakhmi C. Jain, to Dr. Thomas Ditzinger and to the Series *Intelligent Systems Reference Library* staff at Springer for their support in making this book possible.

Andrzej Skowron was partially supported by the grant NN516 077837, from the Ministry of Science and Higher Education of the Republic of Poland, the National Centre for Research and Development (NCBiR) under grant SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

Andrzej Skowron and Zbigniew Suraj

Warszawa, Rzeszów, March 2012

Contents

Chapter 1
Professor Zdzisław Pawlak (1926-2006): Founder of the Polish School of Artificial Intelligence

Andrzej Skowron*, Mihir Kr. Chakraborty, Jerzy Grzymała-Busse, Victor Marek, Sankar K. Pal, James F. Peters, Grzegorz Rozenberg, Dominik Ślęzak, Roman Słowiński, Shusaku Tsumoto, Alicja Wakulicz-Deja, Guoyin Wang, and Wojciech Ziarko

He was not just a great scientist – he was also a great human being.
– Lotfi A. Zadeh, April 2006

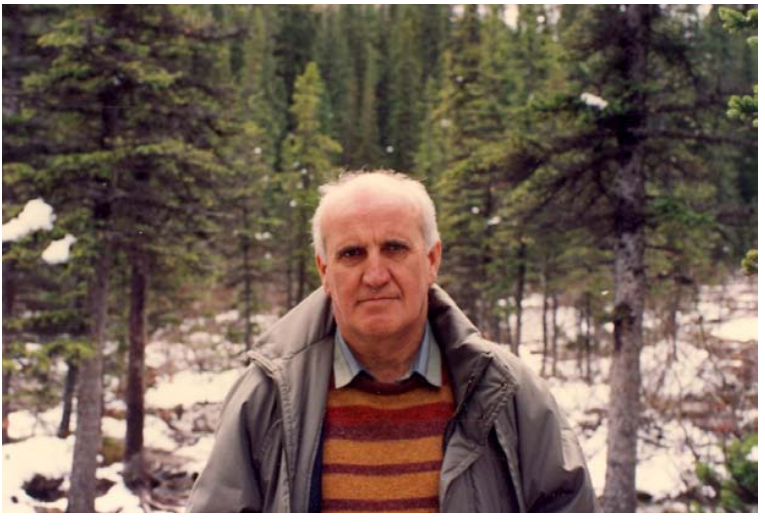


Fig. 1.1. Zdzisław Pawlak

Andrzej Skowron
Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
e-mail: skowron@mimuw.edu.pl

* Corresponding author.

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 1–56.
springerlink.com

1.1 Introduction

This chapter is dedicated to the memory of Professor Zdzisław Pawlak, founder of the Polish school of Artificial Intelligence and one of the pioneers in Computer Engineering and Computer Science with worldwide influence.

To capture the spirit of Professor Pawlak's creative genius, this chapter contains testimonies of many collaborators, colleagues and friends pointing to Professor's scientific achievements and his personal qualities. In short, we present Professor Pawlak as a truly great scientist, teacher and human being.

1.2 Biography [51]

Zdzisław Ignacy Pawlak was born on 10 November 1926 in Łódź, where he also finished primary school in 1939. During the German occupation of Poland, like many Poles, he was a slave-labourer and was forced to work for Siemens. After the Second World War, in 1946, he passed his high school examinations as an extern, and in 1947, he started his studies at the Electrical Engineering Faculty of the Łódź University of Technology. In 1949, he transferred to the Electrical Faculty (the Faculty of Telecommunication between 1951 and 1966, the Faculty of Electronics and Information Technology at present) at the Technical University of Warsaw (now the Warsaw University of Technology). He received his engineering degree in Telecommunications and Master of Science degree in Radio Engineering in 1951, presenting the diploma thesis entitled *A clock for the electronic computing machine*, prepared under supervision of Romuald Marczyński.

After graduation, he worked as a junior member of the research staff at the Mathematical Institute of Polish Academy of Sciences (PAS) (now Institute of Mathematics of Polish Academy of Sciences (PAS)) until 1957. Between 1957–1959 he worked at the Technical University of Warsaw, where he took part in designing the first Polish computer. In effect, one of the first computing machines in Poland was built under his supervision. In 1959, he returned to the Institute of Mathematics of PAS where he worked as an assistant professor from 1959 to 1963.

He received his doctoral degree in 1958 (at the time called *candidate of technical sciences*) at the Institute of Fundamental Technological Research of PAS presenting the doctoral thesis entitled *Application of Graph Theory to the Decoder Synthesis*. The dissertation was supervised by Professor Krystyn Bochenek.

Professor Pawlak received his postdoctoral degree (habilitation, Dr. Sci.) in Mathematics at the Institute of Mathematics of PAS in 1963 for the dissertation entitled *Organization of Address-less Machines*. From 1963 until 1969, he worked at the Institute of Mathematics of the Warsaw University.

In 1971, he was promoted to an Associate Professor at the Institute of Mathematics of PAS. Between 1971 and 1979, Professor Pawlak was the Deputy Director for Science at the Computer Center of PAS, and later, after the institute's renaming in 1976, of the Institute of Computer Science of PAS. In 1978, he was promoted to a

(full) Professor in the Institute of Computer Science of PAS in Warsaw. In 1983, he was elected a corresponding member of PAS, and later, in 1991, he became a full member of the Polish Academy of Sciences. From 1979 to 1986, he was the director of the Institute of Informatics of the Silesian University of Technology. Starting from 1985, he worked at the Institute of Theoretical and Applied Informatics of PAS in Gliwice. In 1998, he also worked at the University of Applied Computer Science and Management in Warsaw. Between 1989 and 1996, he was the director of the Institute of Computer Science of the Warsaw University of Technology.

In 1950, Professor Pawlak developed the first-generation computer GAM-1 at the Group of Computing Machines (GAM) at the State Institute of Mathematics in Warsaw. However, that machine was never used for practical applications. In 1951, Zdzisław Pawlak came up with a new way to generate random numbers, which was published in the prestigious *Mathematical Tables and Other Aids to Computation* journal (now called *Mathematics of Computation*), the oldest journal devoted to computation [17]. It was the first-ever Polish computer science work published abroad. Later, he suggested a new method for representing numbers in the positional numerical system with a negative radix -2 (so-called -2 system). Based on this technique and horizontal microprogramming, with Professor Pawlak's project and supervision, a computing machine UMC-1 was built at the Warsaw University of Technology. Later, Professor Pawlak was studying many aspects of computer science, including computational linguistics, automata theory, automated theorem proving and information retrieval. One of the most interesting achievements of that period was a new formal model of computing machine, different from Turing's machine and Rabin-Scott's finite automata. That model gained a lot of attention worldwide and was called Pawlak machine in the literature. Another important accomplishment was creating the first mathematical model of Crick and Watson's DNA encoding. Pawlak also developed an original approach to the information retrieval. He also proposed a new mathematical approach to the conflict analysis, which has important applications in psychology, economy and politics.

Professor Pawlak's most important discovery was his invention of rough set theory in 1982, which gained vast popularity throughout the World. More than 5000 English-language publications about Pawlak's theory and its applications have been published so far, including several books.

Many international conferences, mainly in the USA, Canada, China, India, Japan and Europe, were organized to discuss and develop Professor Pawlak's work.

At those conferences, he gave lectures, among other subjects, in mathematical logic, mathematical foundations of computer science, organization of computing machines, mathematical linguistics and rough set theory. He was frequently invited as a visiting professor to many universities in the USA, Canada and Europe, including the Philosophy Department at Stanford University (1965).

Professor Pawlak received many honours and awards acknowledging his achievements as one of the main animators of scientific life in Poland. His work was recognized on the national level by the Polish government, including Polish National Science Award in 1973, Polish Knight's Cross of the Order of Polonia Restituta

in 1984, Polish Mathematical Society Steinhaus Prize for achievements in applications of Mathematics for 1989 and Polish Officer's Cross of the Order Polonia Restituta in 1999.

Professor Pawlak was a member and officer of many scientific organizations, active in various periods of time in over 30 governing councils (including being the president of a number of those). In his native Poland, he was the president of the National Central Committee for Scientific Titles and Degrees (CKK) between 1975 and 1988 (mathematical and technical sections) [1], member of the Computer Science Committee of PAS, the Committee of Cooperation of Academies of Sciences of the Socialist Countries on Computational Technology (1971-1979), the State Committee for Scientific Research (1994-2000), the Central Committee for Scientific Titles and Degrees (2000-2006), the Polish Mathematical Society and the Polish Semiotic Society (vice-president, 1990-1996). He served on several editorial boards of scientific journals, both foreign and national. He served as the deputy editor-in-chief of the Bulletin of PAS. On his initiative, the journal Fundamenta Informaticae was created. For many years, he served as the deputy editor-in-chief of Fundamenta Informaticae. He published over two hundred articles and a number of books. Professor Pawlak supervised thirty doctoral dissertations. We quote all these facts to show the amount of his energy and enthusiasm devoted to promotion of scientific research, education of young researchers and their supervision.

Professor Pawlak loved to spend time with family in nature surrounding (see Figure 1.2). So many visitors and friends from all over the world were always very welcome at the home of Professor Pawlak by his wife Danuta.

Fig. 1.2. Professor Zdzisław Pawlak with wife Danuta and daughter Dorota (picture taken in 1950s)

[1] This committee was responsible for the scientific evaluation of candidates applying for D. Sci. (habilitation) degree and Professor title and recommending the final decision.

1.3 From the Clock to the Pseudo-random Number Generator

As it was mentioned before, Zdzisław Pawlak got his engineer's degree in Telecommunications and Master of Science degree (in radio engineering) in 1951, presenting the diploma thesis *A clock to the electronic computing machine*, supervised by Romuald Marczyński. An interesting story is connected with this work, which was told by the Professor at the conference dinner in Jabłonna Palace, near Warsaw, during the First International Conference on Rough Sets and Current Trends in Computing (RSCTC) in 1998 [42] (see the picture below).

Fig. 1.3. Meeting during the dinner at the First International Conference on Rough Sets and Current Trends in Computing (RSCTC 1998) in the Jabłonna Palace (near Warsaw): Professor Zdzisław Pawlak is describing how the clock from his master's thesis was transformed into the pseudo-random number generator

It turned out that the clock designed in his master's thesis, is hardly stable. Then, Zdzisław Pawlak came into the conclusion that the design may be modified and used to create a pseudo-random number generator. All that became an inspiration for the work on flip-flop as a generator of random binary digits, published in 1956 in the *Mathematical Tables and Other Aids to Computation* [17].

Many years later, Professor Pawlak visited one of the American institutes, where he was proudly presented with a fine-quality pseudo-random number generator. He was also told that it was based on the idea of a Polish scientist. When he asked about the name of that scientist, he got the reply, *Pawlak*[2].

[2] See (video) interview with Professor Pawlak at www.atvn.pl/index_sub_page.php?atvn=archiwum/index&title=ARCHIWUM&icm=edit_lista&graf=

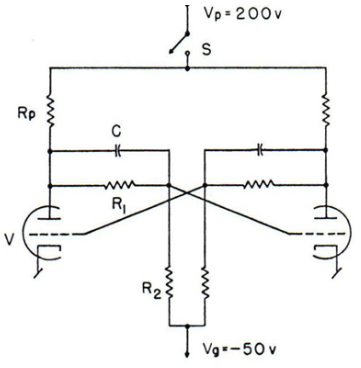


FIG. 1.

In this way we may obtain a finite random series of A and B which are statistically independent. One series produced by the aid of a flip-flop is given below:

AABAABBABBABBBABBAABABABBABAABABABBAA
BABABBBBBBBBBABBBABAABBB.

Let $\{Y_k\}$ be the sequence of k pairs of elements of $\{X_{2k}\}$ such that $Y_i = X_{2i-1}, X_{2i}$, where $1 \leq i \leq k$. Omitting in $\{Y_k\}$ all elements of the form AA and BB we obtain a third sequence whose elements are the pairs AB and BA only, denoted in the following by 0 and 1 respectively.

Let $p_j(\mathrm{A})$ and $p_j(\mathrm{B})$ denote probabilities that j-th switching on of contact S set flip-flop in state A or B respectively and suppose that $p_j(\mathrm{A})$ and $p_j(\mathrm{B})$ are asymmetric, say $p_j(\mathrm{A}) > p_j(\mathrm{B})$. Supposing that the flip-flop does not change its properties during two successive switchings, we may write

(1) $$p_{2i-1}(\mathrm{A}) = p_{2i}(\mathrm{A})$$

(2) $$p_{2i-1}(\mathrm{B}) = p_{2i}(\mathrm{B}).$$

From 1 and 2 we have

(3) $$p_{2i-1}(\mathrm{A}) \cdot p_{2i}(\mathrm{B}) = p_{2i-1}(\mathrm{B}) \cdot p_{2i}(\mathrm{A}).$$

Because

(4) $$p_{2i-1}(\mathrm{A}) \cdot p_{2i}(\mathrm{B}) = p_i(0)$$

Fig. 1.4. Fragment of the work *flip-flop as generator of random binary digits*

1.4 Engineer and Mathematician

As an engineer, Professor Pawlak treated language of Mathematics as a tool to formulate his ideas accurately. He was convinced that Computer Science must be founded on Mathematics. He was one of the pioneers of research direction called

Foundations of Computer Science and initiated a conference with that name. Undoubtedly, his attitude was influenced by years of his work at the Mathematical Institute of PAS in Warsaw. Professor Pawlak's own words, delivered at the Poznań University of Technology, during the ceremony of awarding him a honorary doctoral degree, picture that period best. A short fragment of his speech is presented below [32]:

My first experience of scientific work occurred in 1951 at the State Mathematical Institute (at present, The Institute of Mathematics of PAS), where, after graduating from the Warsaw University of Technology, a group of scientists headed by engineer Romuald Marczyński was building the first Polish computer. This group was called "Mathematical Apparatuses Group" (GAM). Dealing with a totally new area of science, both in Poland and all over the World, I found myself in an unusual situation for a fresh graduate of the University of Technology. At that time, the only existing computer (as we know them now) was located at the Cambridge University. We, the builders, had a complete lack of knowledge, literature and, the most important, skilled 'masters' that would be able to lead our group of inexperienced, young scientists. There was a large group of outstanding, world renown mathematicians working at the Institute of Mathematics of PAS including Professors Karol Borsuk, Wacław Sierpiński, Kazimierz Kuratowski, Andrzej Mostowski, Roman Sikorski, Jerzy Łoś, Stanisław Mazur, Andrzej Grzegorczyk, and others. However, the area of computing machines, which is how computers were then called, was of no interest to them. Work at the Institute of Mathematics gave me an opportunity to interact with many great mathematicians, even from the outside of the Institute, for instance Professors Helena Rasiowa, Kazimierz Ajdukiewicz, Hugo Steinhaus, Klemens Szaniawski, and others. Moreover, thanks to the work at the Institute, I had the opportunity to meet some of the greatest mathematicians of the 20th century, including Alfred Tarski, Stanisław Ulam, Samuel Eilenberg, Alonzo Church, Leon Henkin, Dana Scott, Laszlo Kalmar, Alfréd Rényi, Rózsa Péter, Andriej Kołmogorow, Borys Trachtenbrot, Borys Gniedenko, Andriej Markow, Andriej Tichonow, and others. These contacts had no direct impact on my scientific development but the atmosphere dominating the Institute and daily contacts with eminent scholars indirectly influenced my personal scientific interests a lot. In the case of dealing with our team's specific problems, everybody had to find his own solution. Actually, this had some advantages, because it left us a significant area of independence, but today, I tend to think that disadvantages of that situation overwhelmed its positive sides. It is not the right time to develop the topic, but taking all that into consideration, that situation had significant effects on my personal development as a researcher with both advantages and disadvantages.

After a short period of time, the logician Henryk Greniewski became the head of GAM. He was a polite and kind-hearted man with a great personality. I owe him my first contacts with scientific issues. He was the first one to organise a seminar on Boolean algebras and their usage in synthesis of digital systems, which was a mysterious topic for me at the time. He always had a kind advice for me and payed attention to my problems, not necessarily connected to science. Without hesitation he played a significant role in my scientific development. I organised my first seminar later with my friend Dr Andrzej Ehrenfeucht, an eminent mathematician and a man of many interests, not only limited to mathematics. Our meetings were concerned with computers, algorithms and logic, which definitely broadened both my knowledge and scientific horizons.

1.5 Computation Models, Rough Sets and Artificial Intelligence

It is natural to divide Professor Pawlak's scientific work into a number of stages. The first of these is his work on first Polish digital computer in the 1950s. During this period, Professor Pawlak worked on the organization of digital machines and the logic of operation of digital systems. Coming up with a new method of representing numbers in the positional numeral system with a negative radix (-2 *system*) was a significant achievement of that period. This method was used to implement arithmetic-logic operations in the arithmetic operations unit of an experimental computing machine UMC-1, built under Zdzisław Pawlak's supervision at the Warsaw University of Technology. In the 1960s, a significant Computer Science research was concerned with the logic of digital machines. Automata theory was evolving especially fast. At the time Professor Pawlak was interested in organization of address-less machines, which were controlled by a so-called transition function. Pawlak created a new formal model of an address-less machine, which was different from a Turing machine and Rabin-Scott's finite automata. This proposal created an international interest and was called the *Pawlak machine*. At the time Professor Pawlak investigated computations realized by von Neumann machines and also alternatives, such as address-less machines.

Getting worldwide interest was another obvious success in the initial stage of Pawlak's scientific work. Let us present a few examples.

Professor Pawlak was invited to be a part of an American team, supervised by Professor Traub from University of Illinois, constructing ILLIAC IV, one of the first attempts at a massively parallel computer. Traub was familiar with Pawlak's work since he served as one of the reviewers of the Pawlak habilitation thesis. Unfortunately, the government of People's Republic of Poland did not let him travel, refused to give him a passport and the plan failed.

The idea of the positional numeral system with a negative radix (-2 system) was rediscovered in the USA, ten years after Pawlak's work publication. Let us quote Pawlak's letter [24]:

> *I have recently read the above paper (S. Zohar, IEEE Trans. Computers, vol. C-19, Mar. 1970, pp. 222-226) with interest. The idea of negative radix is not new, and has been dealt with in a series of papers, some of which are noted in the references. We have in Poland over ten computers built on this principle, which have been working for over ten years. This information may be of interest to people working on the subject in the U.S.A.*

Profesor Pawlak, was one of the few Poles invited to make a plenary presentation at the *Congress of Logic, Methodology and Philosophy of Science* in Amsterdam in 1967. The lecture was entitled *On the notion of a computer* [23].

Professor Pawlak proposed a new class of parenthesis-free languages, being a generalization of Łukasiewicz's parenthesis-free notation. These results were included in the fundamental work of D. Knuth *The Art of Programming*.

In the 1970s, Professor Pawlak was interested in the formal models of the DNA. His formal model of genetic codes [21] was the world's first mathematical representation of the DNA. It was the milestone for further research [13], which helped

in incorporating the double helix structure in the formal model, leading to the rapid development of many computation models based on the DNA.

In the sixth chapter of the book Genetic Grammars ([21]), Pawlak introduced a new type of grammars, generating complex systems from the elementary ones, for instance creating proteins from amino-acids. He also presented a generalization of traditional grammars, which is still used in the formal language theory. In the book [21], Pawlak introduced a planar mosaic construction from elementary mosaics using production rules for the composition. He also presented a language for linear representation of such mosaic structures. Pawlak also proposed a two-way approach to grammars, consisting of formal grammars and constructions called picture grammars, see [45, 43, 44, 13]. Professor Solomon Marcus [13] states that Pawlak's research in formal grammars and picture grammars was a pioneering work at the time. Later, the theory of formal grammars was given a full presentation by Arto Salomaa in 1973. The first attempt at creating a general approach to picture grammars was made by Alan C. Shaw in 1967. In 1969, an extensive monograph on the topic was published by Azriel Rosenfeld. Professor Solomon Marcus describes [13] his first contact with this model (presented by Pawlak in a popular science book [21]) and the importance of this event for his own research:

> *41 years ago, Z. Pawlak has published in Polish language a book aimed perhaps as an introduction to the field of mathematical linguistics (Pawlak 1965). Short time after this event, he attended an international conference in Bucharest and I met him there. He offered me a copy of this book. As a matter of fact, he showed me the book and he said that he is sorry to have it in a language which is not available to me. But I told him that I would like to have the book and I will manage to follow it at least partly. Happy idea! Besides some usual introductory notions concerning the mathematical approach to grammars (the title in Polish: "Gramatyka i matematyka" was clearly "Grammar and mathematics"), a special chapter called my attention because it was concerned with the grammar of the genetic code. I was already introduced, at that time, to the works of Roman Jakobson and of many other authors concerning the analogy between linguistics and molecular genetics. Pawlak's approach was mainly presented in symbols, graphs and geometric pictures, while the few words in Polish were in most cases international words like codons, amino acids, nucleotides, proteins.*
>
> *It is interesting to recall the period of the sixties of the past century. After a long period in which historical linguistics used ideas and metaphors of Darwinian biology, an important change took place: instead of using biological ideas and metaphors in linguistics, linguistic ideas and metaphors related to phonemic and morphemic segmentation penetrated the study of nucleic acids, amino acids and proteins.*
>
> *To this itinerary of opposite direction in respect to the previous one, Pawlak was adding the idea of a generative perspective in the study of heredity. In this aim, he proposed a mechanism operating concomitantly in two directions. On the one hand, in the direction of formal grammars, on the other hand, in the direction of what was called later picture grammars.*
>
> *Let us recall that both formal grammars and picture grammars were at that time at their very beginning. Formal grammars theory had to wait till the year 1973 for a first satisfactory rigorous presentation (Salomaa 1973), while picture grammars had to wait until the year 1967 for a first systematic attempt (Shaw 1967) and two more years for the monograph by Rosenfeld (1969).*

> *[...] missing structure [a double helix structure of Watson and Crick in the approach by Pawlak] became the point of departure in Tom Head's pioneering work on DNA computing (Head 1987)*[3].

In the 1970s, Pawlak and his co-workers investigated information retrieval systems (see [25, 14, 15]). In applications, information systems and query languages were expected to provide efficient up-to-date information extraction from the constantly expanding data sets. The formal model of the information system and the language for information retrieval, developed by Professor Pawlak, made it possible to create a unified approach to information retrieval and resulted in a broad analysis of the information retrieval processes' features [47].

Considering information retrieval, Professor Pawlak focused on the connection of his approach with the semantic information [26]:

> *It is worth to think of a more general question, namely: Is, for example, description of data sets possible only in a given way? What exactly is an information? What are its basic features? etc. Let us observe that the notion of information, which we need to achieve the goals presented in this article, has nothing to do with the notion of information formalized in so called theory of information created by Shannon in the 1940s. In that theory, the starting points for describing information refer to sending signals in telecommunication networks. In the 1930s, the logician and philosopher Rudolf Carnap elaborated on the notion of information similar to the one we mention in this article. It might be possible to create a new theory of information based on our experience with the modern computing machines.*
>
> [...] *Almost all problems, which were raised while building and using computing machines have many different aspects - from the ones connected directly with the applications, to the ones that go beyond computing machines themselves. This is probably the reason why computer science - the science describing computing machines and their usage - is so appealing.*

Pawlak's statement about creating new theory of information proved appealing to many scientists. Nowadays, many researchers are engaged in the effort to create such theory. From the first publication on the topic of semantic information [1], the research on the range of problems in semantic information grew immensely, especially in recent years ([5, 2, 7, 8]). There is a need to develop new methods of modelling complex, autonomous and adaptive dynamical systems, in which computations are realized by interactions of their components (information granules). Research on basic notions such as (interactive) computations intensified recently [9, 11].

The above-mentioned model of an information system was used in a number of approaches to data processing and mining, as fundamental representation of available data. Several years ago, the printouts of seminal Pawlak's paper published in 1981 in Information Systems [27] were distributed among the participants of the 28th ACM SIGMOD/PODS conference in Vancouver, as an important complement to Codd's vision of a relational model. This example shows a huge influence of Professor Pawlak's ideas on very diverse areas of science and industry, including in particular the mainstream research on database systems and information retrieval.

The last period of Zdzisław Pawlak's scientific work began in the 1980s and continued until he passed away in 2006. Fundamental works connected with rough set

[3] See [10].

theory are the results of that period [28, 30, 35, 36, 37]. That theory is, undoubtedly, the most important achievement of Professor Pawlak.

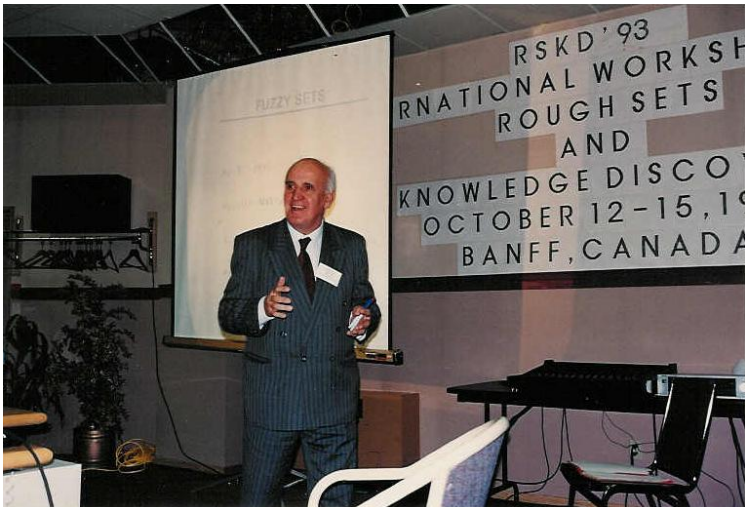


Fig. 1.5. Keynote talk of Professor Pawlak during the RSKD 1993 conference in Banff

Roman Słowiński stated in his laudation for Zdzisław Pawlak's honorary degree in Poznań University of Technology [47]:

> *This theory helps to find answers to many basic questions in mathematics, computer science, artificial intelligence, decision theory, conflict theory, machine learning, knowledge discovery and control theory. This theory is founded on an observation that knowledge about objects from a real or abstract world is granular. Indeed, objects described by the same information are indiscernible and create elementary sets, which are knowledge granules for that world. When willing to express a concept, referring to a given set of objects, in terms of knowledge about the world the objects come from, one encounters a situation in which in general, the concept is not expressible exactly by the available granules; in other words, the union of elementary sets having non-empty intersection with our set, does not coincide with the set. This set - a concept - may, however, be expressed roughly, using sets called lower and upper approximations - lower approximation containing elementary sets (granules) which are wholly included in our set, and upper approximation containing also those sets which are partly included in our set. The difference between those approximations is called a boundary of a set, and contains ambiguous objects, for which one cannot claim with certainty, whether they do or do not belong to our set. Differentiating between definite knowledge represented by lower approximation and approximate knowledge represented by the boundary of a set has a fundamental impact on the deduction process. Rough set theory complements fuzzy set theory and soft computing, with which it now delivers the best tools for reasoning about data bearing different types of "imperfections", such as ambiguity, inaccuracy, inconsistency, incompleteness, and uncertainty.*

Since Pawlak's introduction [28] of rough sets in 1982, more than 5000 English-language publications on this topic appeared in print [56] and over 5000 Chinese-language publications [53] and books (Figure 1.7).

Fig. 1.6. Professor Zdzisław Pawlak receiving Honorary Doctorate from Poznań University of Technology (2002) - Roman Słowiński (standing on the left) delivering his laudation for Zdzisław Pawlak's honorary degree in Poznań University of Technology

Fig. 1.7. Books on rough sets in Chinese [53]

Those publications include both specializations and extensions of rough set theory. Their goal is to solve new scientific problems, examining connections between rough set theory and other approaches to uncertainty and applying the theory in practice. A number of books devoted to rough sets theory were published worldwide.

Numerous conferences, *for example*, in China, India, Japan, Canada, USA and in Europe were organized. For example, in 2011, the Thirteenth International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2011) was organized in Moscow, Russia and the 6th International Conference on Rough Sets and Knowledge Technology (RSKT 2011) was held in Banff, Canada. Many international conferences added rough sets to their lists of principal topics. Rough set theory has an immense following in China. The First Chinese Workshop

on Rough Sets and Soft Computing (CRSSC2001) was organized in May 2001, in Chongqing (see Figure 1.8). Professor Pawlak attended the workshop and gave a keynote talk. Guoyin Wang from Chongqing is reporting that this talk was very important for the development of rough sets in China.

Fig. 1.8. Professor Zdzisław Pawlak (the first row in the middle) with participants of the First Chinese Workshop on Rough Sets and Soft Computing (CRSSC 2001), May 2001, Chongqing

The tenth Chinese conference devoted to rough set theory took place in Chongqing in 2010. In 2010, another international conference named Rough Sets in Knowledge Technology was organised in Beijing, while in Zhejiang, a successive international workshop called Rough Set Theory took place. The 2012 Joint Rough Set Symposium (JRS 2012), that is, a joint conference of the Eighth International Conference on Rough Sets and Current Trends in Computing (RSCTC 2012) and the Seventh International Conference on Rough Sets and Knowledge Technology (RSKT 2012) will be held in Southwest Jiaotong University, Chengdu, China, 17-20 August 2012[4].

Figure 1.9 presents places in China where conferences on rough sets took place.

In his talk during the session devoted to the memory of Professor Pawlak at Rough Sets and Emerging Intelligent Systems Paradigms (RSEISP 2007) conference in 2007, Guoyin Wang stated:

> *Professor Pawlak will eternally remain in the memory of worldwide scientific community thanks to his research achievements, which are his permanent contribution for the World Science. He had a significant impact on building friendship between Polish and Chinese scientists.*

[4] http://sist.swjtu.edu.cn/jrs2012/

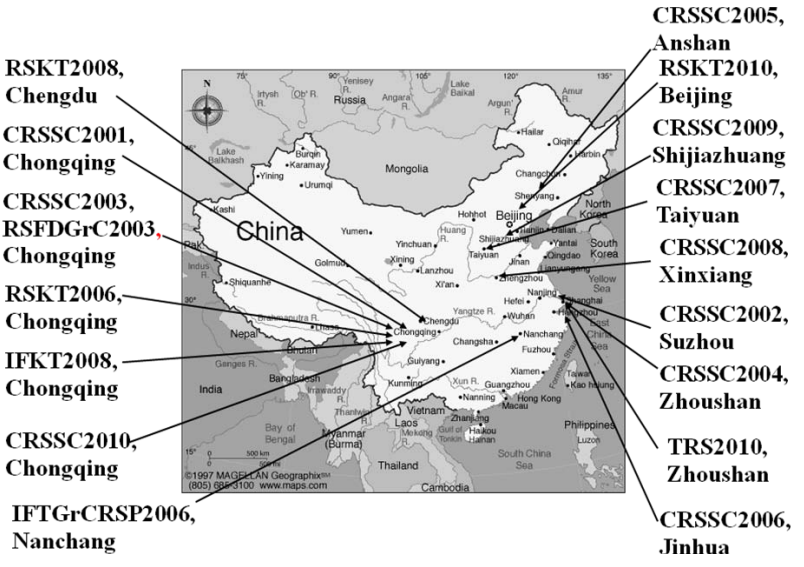


Fig. 1.9. Places in China where conferences on rough set theory were organized [53]

For more information about the research on rough sets in China the reader is referred to [52].

Research concerning rough sets is also significantly growing in India, recently [12]. The year 2009 was called the Rough Set Year in India. There are also a lot of teams working on rough set theory and its applications at many university centres in Poland[5]. Professor Pawlak's book [30] about rough sets was, so far, quoted above 8500 times in Google Scholar. The number of valuable theoretical publications and applications basing on rough sets is constantly growing, especially when combined with other approaches to reasoning based on imperfect (often incomplete) information.

Professor Pawlak is also the author of an approach to conflict analysis [37, 29]. Andrzej Skowron is reporting comments of Professor Pawlak during a meeting in 2006:

> *I remember that during a discussion in 2006,* [*Professor Pawlak*] *was enjoying the fact that his approach to conflict analysis and conflict solving was chosen as a basis for a large project concerning negotiations between shipyards in Hong Kong.*

Out of many awards given to Professor Pawlak, he especially treasured the Hugo Steinhaus Prize awarded to him by the Polish Mathematical Society for his work on applications of mathematics and the honorary doctoral degree of the Poznań University of Technology.

[5] See http://rsds.univ.rzeszow.pl/

In 1996, he received the Lotfi A. Zadeh Best Paper Award. Here is a part of laudation at the award ceremony [55]:

Zdzisław Pawlak, a professor and research scientist at the Institute of Theoretical and Applied Informatics in the Polish Academy of Sciences, has won the 1996 Lotfi A. Zadeh Best Paper Award in the scientific field of soft computing for a paper entitled "On Rough Set Theory".

[...] The award will be presented during the ACM Third Joint Conference on Information Sciences to be held March 2-5, 1997, at the Sheraton Imperial Hotel and Convention Center in Research Triangle Park, N.C..

1.6 Professor Pawlak's Influence on the Development of Computer Science Community

In this section, we will present a number of testimonies of scientists discussing the influence of Pawlak on the development of Computer Science both in his native Poland and in other countries.

Professor Pawlak inspired many computer scientists and mathematicians both in Poland and throughout the world. At present, his students and collaborators head research teams in many countries, including, besides of his native Poland, United States, Canada, Japan, Norway, Sweden and other places. It would be hardly possible to find a computer science institution in his native Poland without encountering faculty influenced by Pawlak. Some research centres, for instance in Warsaw, Poznań, Gdańsk, Katowice, Wrocław and Rzeszów were formed following his initiative. His scientific achievements continue to inspire his many students who are still working there and the next generations of their students. Professor Pawlak had an unusual gift to inspire his interlocutors. As a consequence many individuals were profoundly influenced by his interests and enthusiasm towards scientific research right from the first contact with him [51].

Professor Grzegorz Rozenberg, the Director of the Leiden Center for Natural Computing has written in his letter to the participants of the International Conference on Rough Sets and Intelligent Systems Paradigms (RSEISP 2007, Warsaw, Poland, June 28-30), dedicated to the memory of Professor Pawlak:

I first met Zdzisław since 1963 - he was my mentor, then we became friends, and then family friends: Zdzisław and his wife Danuta became also good friends of my wife and of my parents.

Zdzisław was an unusually gifted scientist whose work is characterized by two main features: genuine originality and elegant simplicity. He had a gift of getting to the real essence, the root, of a research problem. Then he was able to formulate a model that was capturing this root in an elegant and transparent way. His real research interests were always on the boundary of applications and theory: he formulated theoretical models of phenomena that were highly relevant for applications. This reflected well his engineering background. His research was often pioneering - a good example is his model of the structure and functionality of DNA, which he formulated already in 1965.

Fig. 1.10. Professors Zdzisław Pawlak and Grzegorz Rozenberg in front of Mathematical Institute of Polish Academy of Sciences

Zdzisław was an unusually modest scientist - this modesty was totally disproportional to his scientific achievements. He was always amazed that his ideas had such a broad and profound influence. Even when he was describing to me the state-of-the-art of rough set theory, he preferred to talk about the work of others. Usually during such discussions it took me some time to figure out that many of the nice ideas really originated with him. He was a great scientist, certainly the most influential Polish computer scientist. The combination of originality, creativity, and passion for research on the one hand and such a disarming modesty on the other, made him really a role model for scientists.

[...] Zdzisław was a great scientist, but he was also a wonderful person. He had a great sense of humor and a very contiguous laugh - our sessions ended often in attacks of hiccups invoked by telling jokes and writing funny rhymes and poetry.

[...] The essence of what I want to say is that he was a great scientist and a wonderful human being of an exceptional integrity. I was really privileged to have him as a close friend. I surely miss him, I miss our phone conversations, and I often think about him. I

know that many of his friends will remember him for a long time. As a scientist he will be remembered for a very long time as there is no doubt that many of his scientific ideas have a great future.

Victor Marek (University of Kentucky, Lexington, KY) has written in his letter to the participants of the International Conference on Rough Sets and Intelligent Systems Paradigms (RSEISP 2007, Warsaw, Poland, June 28-30), dedicated to the memory of Professor Pawlak:

Fig. 1.11. Professor Zdzisław Pawlak and Victor Marek in Beskidy Mountains during the Workshop on Information Retrieval, 1974

The problem of finding the relevant information is one of the most urgent tasks of Computer Science. Zdzisław was one of the giants who created the theory that underlies the digital revolution. Rough Sets is one of the leading paradigms for thinking about the information, as it is provided to us at the global village through the World-Wide Web.

Sankar K. Pal has written in his letter:

I first came to know about the theory of Rough Sets when I was working at the Software Technology Branch, Information Technology Division, NASA Johnson Space Center, Houston, TX, USA during 1990-92 and 1994 as a NAS-NRC Senior Research Associate. I attended some seminars on applications of rough sets in knowledge encoding in expert systems. I also had an opportunity to take part in discussions of Rough Sets by some of the university researchers who worked in our lab as summer research fellows with my advisor Dr. Robert N. Lea.

As an applied scientist working there at NASA in neuro-fuzzy computing, machine intelligence and genetic algorithms, I got motivated to work on this theory because of its major characteristics like uncertainty analysis, computation with granules and dimensionality reduction. After returning to my Indian Statistical Institute at Calcutta, I took a project from CSIR (Council of Scientific and Industrial Research, India) in 1995 and hired Dr. Mohua Banerjee who had just submitted her PhD thesis at Calcutta University. We started working

primarily on knowledge encoding problems in neural networks for knowledge based connectionist system design, and rough-fuzzy integration for obtaining a paradigm for better uncertainty analysis.

Subsequently, I got involved in an INDO-POLISH collaborative project (funded by DST, India and KBN, Poland) and visited many times Professor Andrzej Skowron and his team at the University of Warsaw. We worked together for different special issues of journals, books, and research papers mainly on granular computing, computing with words and in designing hybrid systems involving rough sets with other soft computing tools. During each of my visits, I met Professor Zdzisław Pawlak, the father of Rough Sets, a very humble and kind man, either at the office of Professor Skowron, or at his office at the Warsaw University of Technology. He attended my talks with appreciation, and we discussed various research issues. I visited his house a couple of times and cherished his paintings and other art. He also visited my Institute once in early 2000 in connection with attending a conference at Calcutta.

Fig. 1.12. In office of Andrzej Skowron (2000); from the left, Professor Zdzisław Pawlak, Marcin Szczuka, Sankar K. Pal

Professor Pawlak is no more, but with his encouragement and blessing we could form a large group in the Machine Intelligence Unit, and establish the Center for Soft Computing Research at ISI, Kolkata, to work with this theory. Currently, the research topics include: granular computing, image processing, data mining, case based reasoning and natural computing involving rough sets either in isolation or in integration with other classical or modern soft computing tools along with their applications in bioinformatics, web mining and video surveillance. Our collaborators in the Hong Kong Poly University and the University of Naples, Italy, are also working in some of these areas. It may be mentioned here that the original definition of Soft Computing of Lotfi Zadeh had four components: Fuzzy Logic, Neuro-Computing, Genetic Algorithms, and Probabilistic Reasoning. Extension of this definition by introducing Rough Sets as Fifth Constituent is the outcome of the aforesaid research contributions of our group. This augmentation enhanced significantly the basic computational intelligence characteristics of soft computing and hence the foundation of the idea and design of high MIQ (Machine IQ) systems.

Recently, we have also formed an Indian Society for Rough Sets to promote rough set research activities in India.

Fig. 1.13. Dinner during the Rough Set Theory and Granular Computing (RSTGC 2001), Matsue, Shimane, Japan, 20-22 May 2001, (from the left) Sankar K. Pal, Professor Zdzisław Pawlak, Andrzej Skowron, Jerzy Grzymała-Busse

While we miss the great man, we love to pay our homage to his memory through the work.

Professor Janusz Sosnowski, the head of the Institute of Computer Science at the Warsaw University of Technology, wrote in his memoir [51]:

After graduation, [Professor Pawlak] worked as a junior member at the Institute of Mathematics of PAS until 1957. In this period, he took part in building the first Polish experimental "mathematical machine" (as computers were named then) GAM-1 [...]. Between 1957 and 1959, he worked at the Warsaw University of Technology, at the Chair and the Section of Telecommunications and Radio Broadcasting (KKTR), which was a very active research and construction projects centre, concerned with digital electronics (called impulse technology then) and "computing machines" [...]. In 1956, a computing machine called PARK (designed by Gerard Kudelski) was built there. Based on the experience gathered during the construction of reliable, complex electronic devices in the 1950s, the PARK group attempted to build its own "mathematical machine." Professor Pawlak played a crucial role in the project. Under his supervision and according to his design, an experimental computing machine was built at the Warsaw University of Technology. Its innovation consisted of using the positional numeral system with a negative radix (-2) and the concept of microinstruction. Based on Professor Pawlak's ideas, an Electronic Digital Machine (EMC) and, later (after Pawlak's transfer to the Mathematical Institute of PAS), a prototype (1960) and five machines of the test series of Universal Digital Machine (UMC-1) were built. The machine [UMC-1] was microprogrammed horizontally, had a drum memory and was realised in the tube technology, based on dynamic digital systems (using Havens' delay lines).

The experience gathered in the UMC-1 project resulted in creating successive generations of digital machines (UMC-10, based on transistor technology). Both, the UMC-10 prototype and the test series of UMC-10, built by the engineers of the Chair of Mathematical Machines Building turned out to be successful and reliable devices. Polish government decided to produce those machines in ELWRO factory (Wrocław, 1962-1964). UMC-10 was the first modern computer produced in Poland. 25 copies of UMC-10 were built and deployed.

Although the period of Professor's work [on UMC] between 1957 and 1959 may appear short, it was of crucial importance for the Warsaw University of Technology, the Faculty and especially for the department which was later transformed into the Institute of Computer Science (in 1975.) It was the beginning of digital and electronic computing techniques

Fig. 1.14. Universal Digital Machine UMC-1

development (the contemporary name of Computer Science). Research and development work [done then] resulted in the long-term development plan of the Department. It also contributed to the fact that the Department was one of the first few centres of Computer Science. Moreover, the Department had significant achievements in educating computer scientists. The first M.Sc students specialising in mathematical machines [i.e. computer engineering] graduated in 1961.

After his transfer from the Department to the Institute of Mathematics, Pawlak's involvement with the personnel of the Department continued. He organised a seminar devoted to problems of contemporary computer science and digital electronics (researchers from the Warsaw University of Technology were also involved in the seminar). Moreover, Professor taught students specialising in building of mathematical machines and worked with many graduates from the Department and the Faculty of Electronic Engineering.

In the 1980s, the political and economical situation in Poland resulted in emigration of many important Institute researchers. Most of them made successful scientific careers in well-known foreign research centres. The result of this exodus was weakening of the scientific staff, especially among researchers with doctoral and advanced degrees. In 1988, the Institute employed only three persons having final (habilitation) degrees. To solve this crisis, we asked Professor Pawlak to come back to the Institute as its head. While being aware of the difficulties that lie ahead, Professor Pawlak accepted the invitation.

For many researchers, Professor Pawlak's return was the additional motivation to work even harder. His experience, authority and international contacts were also very helpful. Pawlak's work allowed to preserve the individuality of the Institute and to define new ways of development. At the time of Professor Pawlak return, the Institute consisted only of the Department of Computer Graphics and six laboratories, responsible for teaching and research. During the Professor's second tenure, the structure of the Institute was changed. Starting in 1994, besides of the Department of Computer Graphics, the Institute created the Department of Software and Computer Architecture and the Department of Information Systems. This decision was crucial for further Institute's development [...].

Based on his knowledge, academic position and authority, Professor Pawlak introduced a new, modern subject, based on a branch of Artificial Intelligence, to the Institute. That was the area of reasoning from uncertain or inaccurate data.

After his retirement in 1996, Professor Pawlak continued to be involved in Institute's matters to the end of his days. He visited the head's office on a regular basis and worked with the researchers of the Institute of Computer Science.

[...] Professor Pawlak had a very important role in integrating the scientific communities of mathematicians and technical scientists. He significantly broadened the research interests of many workers of the Institute. In 1998, the Faculty of Electronics and Information Technology, and the Institute of Computer Science helped to organise an international conference in Warsaw, devoted to rough sets (the 1st International Conference on Rough Sets and Current Trends in Computing (RSCTC)). It was an important scientific event, bringing together experts from many branches of science.

[...] Professor Pawlak was concerned with the pace of computer science development at our Technical University and in the country. He compared it with situation at the similar faculties at the leading foreign universities. He tried to make aware the scientific community of Warsaw University of Technology of the situation.

Professor Piotr Dembiński, director of the Institute of Computer Science of PAS in Warsaw over many years, writes in his memoirs [51]:

[...] As everybody old enough remembers, the beginning of the 1970s and its subsequent years were not the best time for the development of our native computer technology. In that situation, Professor Pawlak decided that it is necessary to focus on the areas in which we were able to succeed worldwide rather than in our country alone. Theoretical (mathematical) foundations of computer science appeared to be such a branch. We – researchers of the Computer Centre – and other people gathered around him were prepared and educated to research this area. Furthermore, the mathematical tools and language were natural to this kind of scientific work.

The choice appeared right. Mathematical Foundations of Computer Science began to shape the scientific profile of the Computer Centre, which was renamed the Institute of Computer Science of PAS in 1976. It is possible to say that Professor Pawlak was an actual creator of the Institute and the main leader of the research conducted there.

To present the results of the Institute's work worldwide, Professor Pawlak motivated us to organise a conference series concerned with the mathematical foundations of computer science in 1972 (Mathematical Foundations of Computer Science - MFCS), which took place annually in Poland and Czechoslovakia (alternating the country each year) and, from 1989, in Poland, Czech Republic and Slovakia. MFCS was the only and unique scientific event of that time, where scientists from both sides of the Iron Curtain gathered. The Conference has not lost its prestige so far.

Professor Pawlak and Professor Rasiowa were also the initiators of establishing a scientific journal called Fundamenta Informaticae, which quickly became renowned in the worldwide scientific community and, as a result, got a high rated position on the ISI Master Journal List.

Alicja Wakulicz-Deja, the president of the Institute of Computer Science at the University of Silesia, recalls Professor [51]:

Professor Zdzisław Pawlak was one of the initiators of creation of the Institute of Computer Science at the University of Silesia and the director of the Institute between 1979 and 1986. He was our co-worker and mentor of the Institute's staff. During his work at the Institute,

he conducted seminars and spent much time on face-to-face scientific discussions with the researchers of the institute.

He always looked for new scientific problems - his ideas inspired his young co-workers. We owe him the first significant publications of the Institute of Computer Science and presentations of our research at many national and international conferences.

Professor Pawlak was a very demanding as a scientist and, while often using quick wit and jokes, he created an atmosphere of great respect for his knowledge and intelligence.

Professor Pawlak left the Institute in 1986 because of his personal and health problems. However, he was always interested in research and work of our Institute. He repeatedly asked me to visit him in Warsaw and report on the progress of the Institute. He was pleased with our achievements and advised how to solve our problems. In particular, he said that "if anybody is able to replace us, we should start researching something new." He also thought that primitive and unfriendly people should be avoided because "if you step in the mud, it will leave a stain on your shoes".

[...] I met Professor Zdzisław Pawlak during my last year of studies at the Faculty of Electronic Engineering (specialization - Digital Machines) in Technical University of Warsaw, when he proposed topics for master theses. I chose the topic called "The Graph Grammars and Digital Mathematical Machines". At that time, Professor Pawlak was a docent (Associate Professor) at the Institute of Mathematical Machines, where he took part in building an arithmetic logic unit for the ZAM 41 machine, which was implemented with Pawlak's system (negative 2 base radix).

Thus my collaboration with Professor Pawlak began from the period of his work, which may be called the phase of "Pawlak Machine" then continuing through the phase of the machine specified with the instruction set (my doctoral thesis) to the information systems (postdoctoral degree) and decision support systems (application of rough set theory).

This collaboration taught me to perceive computer science as a fully mature branch of science, the ultimate goal of Professor Pawlak's. He also thought us that one must not develop a theory which has no applications and frequently told us to search for applications of theories we developed.

Professor Jerzy Dembczyński, Rector of the Poznań University of Technology, wrote in the laudation opening the ceremony of awarding Professor Pawlak a honorary doctorate degree [3]:

Fig. 1.15. The ceremony of awarding Professor Zdzisław Pawlak honorary doctorate degree: Jerzy Dembczyński, Rector of the Poznań University is giving flowers to Professor Pawlak

[...] the time has come to thank one of the pioneers of Polish and international computer science, the hero of this ceremony, much-esteemed laureate, Professor Zdzisław I. Pawlak. His contributions to computer science are respected throughout the world. He was instrumental in creating the first Polish digital machine. He is also the first Pole whose scientific work in computer science was published in a prestigious western scientific journal. Professor Pawlak presented the first mathematical model of the DNA, while the laureate's adress-less machine model earned him great respect throughout scientific community. The abundance of his scientific achievements is still growing; we owe him fundamental work on theory of rough sets and granular computing.

[...] The collaboration of Professor Zdzisław I. Pawlak with the Institute of Automation and, later, the Institute of Computer Science at the Poznań University of Technology started in the 1970s and still continues.

Let us quote Roman Słowiński's words [47]:

Looking at him from our, Poznań, perspective, we are grateful to Professor Pawlak for introducing us to rough set theory at its very beginning and for letting us to perform its first practical verifications - it was the application of rough set theory to decision support in medicine. In 1992, Poznań was honoured to organise the first seminar (see Figure 1.16) dedicated to this subject. This seminar began a series of international conferences in Canada, USA, Japan, and Poland. An international scientific association was also founded – International Rough Set Society.

Fig. 1.16. Participants of the first international seminar on rough sets in Kiekrz near Poznań, 1992; (lying from the left) Professor Zdzisław Pawlak and Barbara Wołyńska, (crouching from the left) Zdzisław Piasta, Cecylia Rauszer, Jerzy W. Grzymała-Busse, Jerzy Stefanowski, Andrzej Skowron, Wojciech Ziarko, Piotr Sapiecha, Leszek Płonka, (standing from the left) Ramin Yasdi, Maciej Kandulski, Mohua Banerjee, Jacek Marciniec, Jerzy Krysiński, Janusz Szymaś, T.Y. Lin, Maciej Modrzejewski, El Sanossy Abobaker Sharif, Lech Polkowski, Ewa Orłowska, Jarosław Stepaniuk, Roman Słowiński, Krzysztof Słowiński

[...] Ladies and gentlemen, we have an honour to reward an eminent scientist, with great achievements for Polish and international computer science development, tireless seeker of answers to difficult and fundamental questions in science, man of impeccable reputation, and Poznań University of Technology's friend of a long standing.

While being an acknowleded authority in scientific research, Professor Pawlak is a humble and kind-hearted man. In contacts with young researchers, he transmits his enthusiasm and friendly encouragement. One can also enjoy talking with him on other topics than computer science - he is a connoisseur of photography, and, moreover, since a couple of years he is painting pictures, which show his extraordinary sensitivity to a subtle play of lights in a landscape[6].

Fig. 1.17. "Winter" painted by Zdzisław Pawlak

Professor Mihir Kr. Chakraborty in his letter "Remembering Professor Pawlak" writes:

I remember my first meeting with professor Z. Pawlak at his drawing room in Warsaw on a chilly winter. We had a long discussion full of excitement on uncertainty, vagueness and his own theory of rough sets. Professor offered me a fantastic dinner. And in the end, I expressed my willingness to work on his theory provided I got a young mind to collaborate. Fortunately, within a short period Mohua (Banerjee), a very talented fresh graduate of the department of Pure mathematics, University of Calcutta approached me to supervise her Ph.D. work. I asked her to read some papers on rough set theory and to see if she liked them. Fortunately again, Mohua took up the subject with great enthusiasm and completed her dissertation. She was the first researcher in this field in India. Since that time we have been working together. Jointly we wrote a paper with the title "Pawlak's landscaping with rough sets" – reflecting that Pawlak was a painter who used to paint with rough brush-strokes. We are trying to popularize the theory in India and feel that some amount of success has been achieved. Mohua also was very close to Professor Pawlak. When we received the news of his demise we had been greatly shocked and sent a few lines to Professor Andrzej Skowron as a tribute to Professor Pawlak and as an expression of our deepest respect to his memory,

[6] See Figure 1.17.

I would like to present those lines once again.

Good bye Professor Pawlak

What had been inevitable has happened.
We had been silently apprehending this last moment.

We cherish all the nice memories of every bit of our togetherness
the evenings, nights, working mornings, cosy corners, the lody
riversides, long drives, smiles, quarrels, Warsaw, Paris, Banff
the Niagra, Calcutta

We wished to meet you in pains. We could not

Now we shall paint you on our heart as you painted the nature on canvas
We could roughly walk along the beautiful path you have created as a mathematical-artist
We hope to go on further

Our regards, tears and best wishes for your steps into the unknown

Good bye Papa Pawlak.

Fig. 1.18. Visit of Mohua Banerjee in Warsaw 1992, (from the left) Professor Zdzisław Pawlak, Mohua Banerjee and Janusz Kacprzyk

Fig. 1.19. During International Conference on Fuzzy Systems (AFSS 2002), Calcutta, India, 3-6 February 2002; Professor Zdzisław Pawlak with parents of Mohua Benerjee and Mihir Kr. Chakraborty (on the left)

Shusaku Tsumoto in his letter “Encounter with Professor Pawlak” writes:

Fig. 1.20. RSFDGrC 1999, Yamaguchi, Japan; (from the right) Professor Pawlak, Shusaku Tsumoto with wife and Ning Zhong

My Research Concern from 1986 to 1988

I was involved in development of RHINOS, and interested in characterization of medical reasoning when I interviewed the domain experts for knowledge acquisition. On the other hand, development of electronic patient records was ongoing, and I forsaw that healthcare records will be used as a dataset and automated acquisition of knowledge base would be an important topic in medical informatics.

Residents from 1989 to 1991

After I was graduated from Osaka University, School of Medicine, I worked for Chiba University Hospital as a resident of Neurology from 1989 to Sep 1990, and moved to Emergency Department of Matsudo Municipal Hospital for one year. From Oct 1991, I worked as a research associate of division of medical informatics, and was involved in development of hospital information systems.

Encounter with Pawlak’s book: 1991

When I worked for emergency department of Matsudo municipal hospital, every week I visited a large bookstore called Yaesu Book Center in front of Tokyo Station. I found Pawlak’s book on one Saturday afternoon. I looked it through and found it interesting. However, I was looking for other book on AI and I hesitated buying it. I repeated taking it in hand and putting back into the bookshelf. Since I did not find any interesting book that day, I bought his book. At that time, I did not notice that reasoning of our expert system corresponds to rough sets and I put Pawlak’s book on the desk, not the bookshelf.

Encounter with Rough Sets: 1992 to 1993.

I read his book during Xmas in 1992 in order to give some subject to the seminar in Tokyo Medical and Dental University. In 1993 when I moved to this university as an associate professor, I had found that the core ideas have correspondence to the reasoning of our expert system (RHINOS). That is, exclusive rules and inclusive rules are corresponding to the rules obtained from upper and lower approximation of a given concept. I started my study on rough sets since then.

Workshop in Banff: 1993.

I found a name "RSKD2003" (Rough Sets and Knowledge Discovery) in IEEE Expert and I sent an email to Professor Ziarko. Although the deadline had passed, he encouraged me to submit my papers. I presented my idea in RSKD2003, which was a very nice workshop and I met Pawlak's family: many important persons such as Professors Pawlak, Skowron, Słowiński, Ziarko, Cercone, Lin, Yao and Hu. This was the starting point of my research on rough sets and data mining.

Meeting with Professor Pawlak

I attended Pawlak's plenary talk on conflict analysis, which can be viewed as formal analysis on asymmetric relation [31, 34]. This was the most impressive talk in this workshop: Professor Pawlak was very strong, energetic and research-minded. This is the moment when I had decided to work for rough sets.

Here is a recollection of Wojciech Ziarko about Professor Pawlak visits in Regina:

Professor Zdzisław Pawlak visited Regina several times, about five or six times, in 1980's, early 1990's and early 2000 . I find it difficult to determine precise dates and lengths of his stay at this time since all my office documents are temporarily in storage and not accessible. Prior to my arrival to Regina in 1982, during my teaching contract in Nigeria, we maintained regular contact by letters. Relatively soon after starting my position at the University of Regina, I invited Proessor Pawlak to officially visit our University. It was 1983 or 1984. During this first visit, Professor Pawlak stayed in the city's oldest hotel (considered the most prestigious) called Hotel Saskatchewan. I remember, first morning after Professor Pawlak's arrival, I went to the hotel to pick him up. I met him in the lobby and found him very amused. It turned out that few minutes ago he met in the hotel lobby the editor from Poland of his most recent book, whom he met a day or two ago in Warsaw!

For the first visit, Professor Pawlak brought with him a bunch of manuscripts about his new research. The manuscripts were about set approximations, which he called rough sets. He made a seminar presentation about his research, which was very well received, despite the fact that probably many people in the audience did not quite grasp the basic ideas at that time, me included. It took us many discussions and arguments between three of us, that is, Professor Pawlak, Dr. Michael Wang and myself before things clarified. This was the beginnings of the rough set research at the University of Regina.

After Professor Pawlak left, Michael Wong and myself caught the "rough set bug" and worked together on this new exciting topic, often arguing noisily on corridors. We were, in particular, interested in extensions of the basic theory and its probabilistic aspects. During his next visit to Regina in late 1980's, Professor Pawlak joined our discussions and was very active participant. Some of the results of the discussions are summarized in our joint article [54].

The longest visit of Professor Pawlak to Regina took place also in late 1980's or early 1990's. Professor Pawlak taught a graduate-level class about rough sets at our University, which attracted large number of students. He received very good reviews after this class, which inspired some well-known researchers today, such as Dr. Y. Yao, to focus on this area. He also offered an industrial course on rough sets to the employees of Westbridge Corporation, the largest IT company in Saskatchewan at that time. Professor Pawlak spent about six months in Regina during that stay. He lived in our house, which gave us plenty of opportunities to discuss research, and also to get to know each other well. Professor Pawlak's friendly and brilliant personality was a magnet to our friends and our kids alike, who treated him as second dad. It was also difficult time for Professor Pawlak since his grandson was born when he was here and by necessity he was separated from him for many months. All members of my family have very fond memories of Professor Pawlak's time with us.

Other visits to Regina were relatively brief, connected with two rough set conferences in Banff, in 1993 and 2000. Due to his poor health, Professor Pawlak was not able to attend the rough conference in Regina, in 2005. However, he was in the mind of every participant as he is remembered by all of us who had the privilege of knowing him personally and working with him.

Dr Urszula Stańczyk, in her memoirs about Professor Adam Mrózek from the Institute of Theoretical and Applied Informatics of PAS and the Silesian University of Technology in Gliwice, writes [49]:

In the beginning of the 1980s, Professor Mrózek contacted Professor Zdzisław Pawlak, the creator of rough set theory. Elements of this theory turned out very quickly to be effective in analysis and minimization of arrays describing behaviour of operator-experts. Meeting Professor Pawlak had a crucial impact on A. Mrózek's future research. Applying rough set theory, in the last years of Professor Mrózek's life, resulted in introducing the idea of so called rough controller and broadening the fields of theory's application to economical processes and medical diagnosis.

The last example of Professor Pawlak's inspiration outlined in this section, reported by Dominik Ślęzak, relates to development of a commercial database system available worldwide since 2005 (with its open source version launched in 2008[7]). The idea behind its efficiency lies in data granulation and adoption of principles of rough set approximations to dynamically identify only those granules which are truly required to resolve the incoming SQL statements. Let us refer to two fragments describing this technology, taken from the blog of one of the most influential database analysts[8]. The first fragment was published in 2008:

[...] The "rough set" part of Infobright's story is a lot of mumbo-jumbo [...]

The second fragment comes in 2011:

[...] Rough Query estimates query results [...] To me, Rough Query is the most impressive part of the Infobright 4.0 announcement. [...]

[7] www.infobright.org

[8] www.dbms2.com

In our opinion, the two above excerpts illustrate that it always takes time for industry to accept new ideas. Moreover, only those ideas which are simple and powerful enough can survive. It cannot be a coincidence that so many ideas of Zdzisław Pawlak are nowadays present in so many areas of research and applications.

We close this section with some pictures from the rough set meetings.

Fig. 1.21. San Jose 1994: International Workshop on Rough Sets and Soft Computing (RSSC 1994), San Jose, CA, USA, 10-12 November 1994; from the left: Krzysztof Słowiński, Lech Polkowski, Andrzej Skowron, Robert Golan, Marzena Kryszkiewicz, Jerzy Grzymała-Busse, Shusaku Tsumoto, Zdzisław Piasta, Krzysztof Krawiec, Wojciech Ziarko, Roman Słowiński, Professor Zdzisaw Pawlak

Fig. 1.22. Joint Conference on Information Science, Wrightsville Beach, 1995; from the right: Professor Zdzisław Pawlak, Marzena Kryszkiewicz, Hung Son Nguyen, Jerzy Grzymała-Busse, Wojciech Ziarko, Anna Buczak, Zbigniew Raś

Fig. 1.23. Tokyo 1996: Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery (RSFD 1996), Tokyo, Nov. 6-8, 1996; from the left: Jerzy Stefanowski, Shusaku Tsumoto, Lotfi Zadeh, Wojciech Ziarko, Professor Zdzisław Pawlak, Roman Słowiński

Fig. 1.24. Visit in Kyoto after RSFDGrC 1999 conference in Yamaguchi, Japan; from the right (in front): wife of Grzymała-Busse, Sheela Ramanna, from the right (behind) James Peters, Marcin Szczuka, Jan Żytkow, Andrzej Skowron, Professor Zdzisław Pawlak and (behind) Yoshitsugu Kakemoto

1.7 Zdzisław Pawlak and Artificial Intelligence

In this section, we discuss the idea of Artificial Intelligence as developed in the work of Professor Pawlak and his closest collaborators, including Professor Helena Rasiowa.

Here is a recollection of Andrzej Skowron.

Professor Pawlak and Professor Helena Rasiowa conducted a research seminar on automated theorem proving, at the Faculty of Mathematics and Mechanics of the University of Warsaw.

Fig. 1.25. Professor Helena Rasiowa and Professor Zdzisław Pawlak

I remember this as if it was today. I remember a big auditorium filled with participants of the seminar. In that period, Professor Pawlak conducted an intensive research related to mathematical models of computers and computations realised by them. He lectured and conducted seminars for students of mathematics and computer science. The cooperation with Professor Helena Rasiowa and her research team began in the early 1960s and lasted for many years. The results of this collaboration are still important. One may safely say that a new research school came into existence [11]. This cooperation had a lot of influence

on shaping many people's scientific research and on the evolution of main notions of logic researched by Professor Rasiowa's group [11]: from the classical logic to non-classical logic and its inference processes, the main characteristics of current Artificial Intelligence investigations.

We found the first trace of Professor Pawlak's interest in Artificial Intelligence in an article [18] from 1956, in which he discusses the relationship between mathematical machines ("apparatuses") and cybernetics:

Robots, humans, homunculuses, mechanical animals, chess machines, and other similar wonders - until recently the domain of mad scientists, inventors, alchemists, common deceivers and, at the best case, science fiction writers - are now getting popular among absolutely normal, lucid people and even renowned and respected scholars. This change was triggered during the last war by a famous American mathematician Norbert Wiener. [...] He came into conclusion that modern mathematical apparatuses are almost ideal models of many phenomena happening in the nervous system and, partly, models of phenomena happening in a society. This created basis for renaissance of mechanicism in biology and sociology. This new mechanicism is called cybernetics.

Professor Pawlak warned in [18] that

[...] it is worth to be aware of the fact that an apparatus is not an organism and the analogies between them are secondary, while differences are fundamental.

and

[...] Capabilities of existing, even the strongest apparatuses do not exceed a range of "primitive work".

[...] Looking at the "electronic brain" [...] a faint sign of intelligence and thinking should be expected - features, which no mathematical apparatus has. [at present]

Similar opinions appear again in [19] in 1963:

The role of mathematical machines in mathematics has its strictly determined limits. It seems even less probable for them to play a significant role in other sciences, especially in the humanities. The hopes of cyberneticians for creating a homunculus seem unfounded. [...] there is a similarity between machines and living organisms but it has a surface character, while the differences are fundamental. The history of machines differs from the one of bio-organisms and I do not believe they are ever going to converge.

Is cybernetics a worthless game, then? Probably not. Cybernetics extended the range of engineers' interests to the humanities, while humanists' to engineering. To sum up, cybernetics may indirectly influence the way of the technical development, as well as other sciences progress.

In a later period, up to the beginning of the 1980s, Professor Pawlak did not publish works which were directly concerned with Artificial Intelligence. However, his works, mentioned above, referring, *for example*, to computation models realised by address-less machines [20, 22] or inspired by biological processes [21, 13] clearly shows that he was intrigued with various computation models and searching for alternative models to those realised in von Neumann's machines, models that had the potential to cause the next technological revolution. He expressed that idea in the lecture during the award ceremony at Poznań University of Technology [33]:

We are still unable to provide the parallel algorithm theory, in spite of the huge development of parallel and concurrent systems. New computation models are being developed for, e.g., *DNA computing and quantum computing. In this context, it is instructive to think about Noble Prizes awarded in 1998 for the work related to computers:*

- *in Physics, for the results in researching quantum phenomena as a basis for computers (Robert Laughlin, Horest Stoermer, Daniel Tsui);*
- *in Chemistry, for the development of computing methods (Walter Kohn, John Pople).*

[...] meteorology, aerodynamics, genetics or cryptography demand significantly bigger computational power. I refer not only to increasing the speed of computing but, generally, to finding a new computing paradigm, because modern computers, based on von Neumann's idea, are reaching the limits of their capabilities. This task may be very difficult to realise without new concepts of concurrent and parallel computing on a large scale.

As we see, Professor Pawlak was still intrigued by new computation models. He searched for them, while working on computation models of von Neumann's machines, computation models inspired by biological processes or computation models for automated theorem proving. However, he found all of these models insufficient to solve the real difficulties in creating Artificial Intelligence [33]:

In spite of the computers' enormous successes in science, their role is limited. In the most important scientific tasks: creating and verifying hypotheses, computers did not play any important role so far. The example of Fermat's Last Theorem is symptomatic here. This is because we do not understand the essence of a scientific discovery and the role of intuition, associations etc. Picasso commented on this in a very dramatic manner. (Compare with the motto of [33]: "Computers are useless. They do not pose questions.")

We quote another excerpt of Professor Pawlak's lecture at the Poznań University of Technology [33], referring to issues connected to computation models and Artificial Intelligence:

Proving a hypothesis in inductive logic is done, unlike in deductive logic, not by formal reasoning, but on the basis of experiment. Physics illustrates it best. Researching inductive logic has a long history. An eminent English philosopher John Stuart Mill (1806-1873) is considered its founder. The creation of computers and their innovative applications influenced the rapid growth of interest in inductive reasoning. At present this area develops really dynamically thanks to computer science techniques. Machine learning, knowledge discovery, reasoning from data, expert systems, and other techniques are the examples on new directions in inductive reasoning.

[...] A research in the theory of induction owes computer science new impulses too. However, we are far from the situation similar to the one in deductive logic. There are no emerging outlines of theory of induction, having the same status as the theory of deduction.

[...] Finally, the most interesting technology, from the computer scientist's point of view, is the common-sense reasoning. This is the reasoning which people use in everyday life, politics and many humanities. The starting point of that kind of reasoning is knowledge possessed by a certain group of people ("common knowledge") about a subject, and intuitive methods of reasoning from that knowledge. Examples of that type of reasoning are commonly seen in press, radio and television. They are concerned with politics, economy or arts. Parliamentary debates are a classic example of common-sense reasoning. Governing

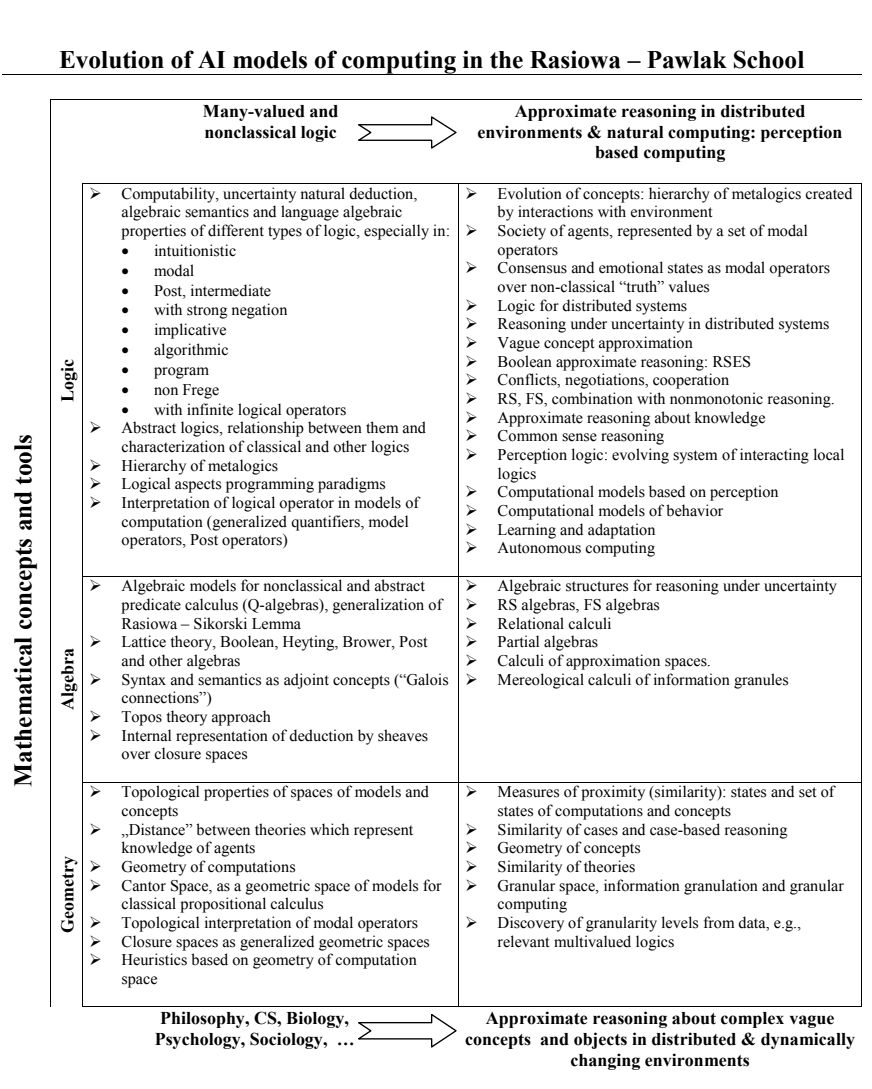


Fig. 1.26. Models' evolution in the Rasiowa - Pawlak research school

parties give arguments for accepting the budget, saying that it is excellent, while opposition parties counter their arguments. Who is right, then? Neither deductive logic (reasoning) nor inductive logic (experiment) may solve the problem. That is why voting is the only chance to settle the argument. The result of voting does not make a thesis valid or not. Of course this method is unacceptable in mathematics or physics. Nobody will judge the correctness of Fermat's theorem or Newton's equations by voting. The theory of this type of reasoning

still awaits significant development. Moreover, its structure is not sufficiently understood. However, there were some investigations of the subject. The common-sense reasoning is really significant because of its range and importance in some fields. Computer science may play an important role here, on the condition that one would understand the essence of this reasoning thoroughly. This can be achieved by making an appropriate theoretical studies.

Undoubtedly, the discovery of rough set theory gave a strong impulse to work on understanding the essence of reasoning mentioned above. Some investigations point to the connections between rough sets and existing formalisms of common-sense reasoning including default logic [4]. Research on relationships between rough sets and paraconsistent logics still flourishes [50]. In our opinion, further understanding of common-sense reasoning's essence may be achieved with better understanding of the fundamentals and capabilities of interactive granular computing. Rough sets play a significant role in defining this form of computing. The programme of this research is presented in [11].

We close this section with a statement of Professor Toshinori Munakata (Cleveland State University) [16]:

It is an honor to contribute my short article to this special issue commemorating the life and work of Professor Zdzisław Pawlak. In this article I would like to discuss my encounters with the field of artificial intelligence (AI) in general, and how I see rough set theory and Professor Zdzisław Pawlak in this context. I have been fortunate to know some of the greatest scholars in the AI field. There are many of them, but if I had to choose the three I admire most, they are: Professors Pawlak, Lotfi Zadeh and Herbert A. Simon. There are common characteristics among all of them. Although they are the most prominent of scholars, all are frank and easy and pleasant to talk with. All are professionally active at ages where ordinary people would have long since retired.

[...] For knowledge discovery techniques such as rough sets, there may be a limit when we deal only with decision tables. Perhaps we should also look at other formats of data as well as other types of data, for example, non-text, comprehensive types of information, such as symbolic, visual, audio, etc. Also, the use of huge background knowledge, in a manner similar to human thought, would be necessary and effective. Human-computer interactions would also enhance the discovery processes. Other totally different domains are non-silicon based new computing paradigms. I am currently working on my fourth Special Section for the Communications of the ACM as a guest editor on this subject [...]. These approaches may lead to a new dimension of information processing in a wide range of application domains including rough sets. As with other scientific developments in history, such as alchemy and the first airplane, a breakthrough may come in a totally unexpected form.

1.8 People and Nature

Professor Pawlak was a lively and witty person. He felt comfortable among other people (especially friends, see pictures included in this chapter 1) and he adored nature. After relaxing surrounded by nature he would regenerate fast and get back to his intensive scientific work, for which he dedicated his life.

Fig. 1.27. Visit in Kansas, Professor Zdzisław Pawlak enjoying activities on slide with children of Jerzy Grzymała-Busse; from the top: son Jan, Professor Pawlak, daughter Anna, and son Witold

Fig. 1.28. Visit in Kansas, Professor Zdzisław Pawlak in a military uniform with Jerzy Grzymała-Busse family; from the left: son Jan; wife Dobroslawa; daughter, Anna; and Jerzy Grzymała-Busse

Jerzy Grzymała-Busse writes about visits of Professor Pawlak at the University of Kansas:

> *Between 1984 and 1994 Professor Zdzisław Pawlak visited the University of Kansas several times. The main reason for his visits was to present his research on rough sets, mostly through invited scientific talks. During these visits, Professor Pawlak lived in the home of Jerzy Grzymałla-Busse and became - practically - a family member. They went on many walks together, as well as trips to local attractions. He participated in various family activities, including the kids' track and field competitions, and trips to a shooting range to shoot M14 and M16 military rifles.*

He loved to go with the younger members of the Grzymała-Busse family to playgrounds, participating and enjoying the activities (Figure 1.27). One of his hobbies was hunting for interesting and rare things at local garage sales. Figure 1.28 shows Professor Pawlak proudly sporting a military uniform.

James Peters (University of Manitoba) writes:

In a short time that I knew him, he was like a father for me. ... He changed my [life] in many ways.

Professor Pawlak was not only a pioneer in mathematics and computer science, he was also a gifted painter of scenes from nature. His interest in enshrining moments in the changing seasons in the Polish countryside in paintings began in the 1950. One of his earliest paintings (a waterscape) was signed by him in 1954. His interest in painting continued through his life.

His paintings capture various moments in the four seasons in the Polish countryside. In a vivid way, his paintings offer his perception of the symmetries and singular beauty of the woods, gently rolling terrain, lakes and shorelines that he visited in Poland. A persistent theme in Professor Pawlak's paintings are watershadows of trees and waterway reeds. These watershadows manifest Professor Pawlak's interest in the portrayal of actual objects such as trees in an approximate but beautiful way. In some sense, Professor Pawlak painted border regions of elongated watershadows that is reminiscent of his basic idea of set approximations in rough set theory. See, e.g., *Figure 1.29 for a sample painting by Professor Pawlak from the late 1990s.*

Fig. 1.29. A sample painting by Professor Pawlak from the late 1990s

Professors Andrzej Czyżewski and Bożena Kostek describe a story behind the Alaskan scenery:

In 1998, May 4th-9th Andrzej Czyżewski and Bożena Kostek had an occasion of participating in the 7th IEEE International Conference on Fuzzy Systems organized in Anchorage, Alaska, USA. A special session devoted to rough sets was organized at this conference, thus it attracted some members of the rough set community, including the founder of rough set theory - Professor Zdzisław Pawlak, who presented a keynote speech at that conference.

On May 10th, the day following the conference, they organized a car tour around Anchorage in order to catch a glimpse of the incredible Alaskan scenery and landscapes. Professor Pawlak and Lech Polkowski accepted their invitation and they all had a very nice time together, because the previously capricious weather changed on that day allowing us to watch the sun beaming to snowy mountains (see Figures 1.40-1.41).

Professor Pawlak seemed impressed with the picturesque view of the snowy peaks and wild landscapes. As it was revealed later, those impressions lasted for long in his sensitive painter's soul. When Andrzej Czyżewski went to Professor Pawlak's Warsaw home many years later, he asked for visiting the small painting gallery located in a room of the home. Among many Professor Pawlak's paintings the Alaskan landscapes shone brightly, catching an attention of the viewer. The last visit of Andrzej Czyżewski in Professor Pawlak's home located at Zuga street in Warsaw was tinted with unspoken awareness of the serious illness of the host. Despite that, as usually, Professor Pawlak seemed to take it easily and even joked. He said: "take this picture now, because after painter's death the price may rise seriously". With those words he took the painting off from the wall and handed it to Andrzej Czyżewski, who got a little bit confused with this act. "I cannot accept such a gift" - he said. Professor Pawlak responded: "it is not for free, you have to pay for it, but not for me. Pay one million dollars or less (he smiled jokingly) to some people who may really need money". Soon they parted, exchanging their last view when Andrzej Czyżewski loaded the painting into his car. He decided right after to respect the donator's will and supplied a hospice in Poland in return to Professor Pawlak's gift.

Today the painting (see the photocopy of it in Figure 1.30) hangs on the wall inside Andrzej Czyżewski's and Bożena Kostek's home reminding them Professor Zdzisław Pawlak who was world-renowned scientist, with so many talents and at the same time so unpretentious, approaching much younger colleagues in a very natural, friendly and warm manners.

Fig. 1.30. A sample painting by Professor Pawlak: Alaska, 1999

Professor Dembiński (Institute of Computer Science of PAS) writes in his memoirs [51]:

> *Professor Pawlak was a modest person, and often began his presentations with the words "I am not sure, if it is significant, but [...]*
>
> *[...] Professor Pawlak had wide interests and had many talents, for instance he sang very well and he knew probably all operetta arias by heart. At the end of his life he was often painting. His paintings could compete with the work of professional artists. He often jokingly collected diplomas attesting to his various achievements, not only scientific ones. So, for instance he got from my wife - a professional dancer - certificate about his high dance skills. Similarly, in pilot school ("Szkoła Orląt") in Dęblin, he asked commanding officer - General Olszewski - for a diploma of a jet plane test pilot, after his flight with General.*
>
> *[...] Professor Pawlak – outstanding scientist and mentor – a person with many talents.*

Professor Janusz Sosnowski of the Warsaw University of Technology describes Professor Pawlak with these words [51]:

> *He was a very cheerful person with a sense of humour and abundant interests beyond the scientific ones. Taking over the Institute's administration after Professor Pawlak's retirement (1996), I was facing many difficult problems both inside of the Institute and in our relations with the administration. At that time, helpful discussions with the Professor assisted me in surviving difficult moments and finding appropriate solutions. Professor Pawlak was also able to appreciate achievements of others and willing to help them in their work, which are unusual features in scientific environment.*

Mr Jerzy Fiett, Professor's friend, writes in his testimonial presented during one of the sessions devoted to the memory of Professor Pawlak:

> *How many years have passed, when young (then) people: Zdzisław Pawlak - "Kłaczek", Mieczysław Zielczyński - "Miećka", Andrzej Janikowski and Jerzy Fiett, members of exclusive Old Bachelors' Club from the 1950s, made up a group that could do the impossible and put even the wildest ideas into operation? Andrew is not with us anymore. He rests eternally in Stara Miłosna. Zdzisław, after achieving everything in his scientific career, still works in his profession, in spite of serious health problems. Moreover, he paints really great, writes, and still has new plans.*

Professor Pawlak often recalled excursions to Bieszczady Mountains, which took place after the Second World War. Here is a fragment of Mr Fiett's memories of these trips:

> *A camp on Szeroki Wierch, frost cracking of trees, a trip to Halicz via Tarnica - Krzemień (that damn frost traverse!), return at sunset, beautiful downhill ride from Tarnica on the snowy drifts. Another freezing night, drying and freezing boots for the night, drying gloves and socks with "on belly" method - beautiful morning, hoarfrosty replicas of socks and gloves on our sleeping bags! Stonefrost, icy boots. Next, after a proper breakfast, to the place, where Berehy Górne used to be. Incredibly hard struggling to go with resistant toboggan, constantly playing tricks, through the snowy, sloping bushes, filled with surprises in the form of snowdrifts, rifts and other obstacles. Next, walking on a bit friendlier terrain for some time, to the tiny Lemken cemetery and a few chimneys sticking out from the snow. That was all that was left from Berehy Górne village. Near the site of a fire, on a broad*

opening covered with deep snow, frozen to the bones, we put up our tents and prepare a place to sleep. We collect timber to make a bonfire and surround our tents with previously cut wood. That is how we made our new campsite. Sleeping is not going to be comfortable, like during the previous nights. We cover the floors of our tents with cloth, on which we lay a self-made mattress, consisting of four rubber tubes and sackcloth binding. We lay our blankets and sleeping bags on the improvised floor and use backpacks as pillows.

Fig. 1.31. Picture made by Professor Pawlak in the early 1950s in Bieszczady

Andrzej Skowron recollects:

Masurian lakes were Professor's most favourite areas, and we used to go there together during the last years of his life. I had an impression that not only does he know every lake there, but also every path, stone and tree.

Here is another fragment of Mr Fiett's memories of trips to the lakes:

The day after a relatively peaceful canoeing trip's segment, the most beautiful but the wildest and the most difficult section (called "a little hell") begins. We row through the great tunnel, shaded with thick branches of trees growing on both sides of the river, going with the strong current. We row through the winding riverbed swarming with rocks, sometimes pretty sharp, and with fallen trees, struggling to protect our boats from breaking - as we can, using paddles or hands. However, it is really beautiful here! Zdzisiek [Professor Pawlak's nickname]*, intrepid photographer, takes photos of his companions (Andrzej and Miećka). These photos were a precious memory of our trip down Wel river and were awarded first prize in the [London] Times' photographic contest (see Figure 1.32).*

Fig. 1.32. Polish jungle: Times' photographical contest award (1950s)

Fig. 1.33. Professor Pawlak with (from the right) Boris A. Trakhtenbrot, Cecylia Rauszer, wife of Professor Trakhtenbrot and Helena Rasiowa

Fig. 1.34. Professors Zdzisław Pawlak and Lotfi A. Zadeh

Fig. 1.35. Charlotte, 1984; from the left: Professor Pawlak, Viktor W. Marek, Elizabeth Marek, Natalia Marek, Professor Rasiowa

Fig. 1.36. Professor Zdzisław Pawlak and Hiroakira Ono with wife

Fig. 1.37. Professor Pawlak in U.S.A. with (from the right) Anita Wasilewska, Zbigniew W. Raś, Shusaku Tsumoto, Wojciech Ziarko

Fig. 1.38. With Setsuo Ohsuga: Birthday of Professor Pawlak in Japan

Fig. 1.39. RSCTC 1998

Fig. 1.40. From trip after FUZZ-IEEE 1998, Anchorage, Alaska; (from the left) Andrzej Czyżewski, Lech Polkowski, Professor Zdzisław Pawlak

Fig. 1.41. From trip after FUZZ-IEEE 1998, Anchorage, Alaska; (from the left) Professor Zdzisław Pawlak, Bożena Kostek, Lech Polkowski

Fig. 1.42. On the way to Banff, RSCTC 2000; (from the left) Professor Zdzisław Pawlak, Jerzy Grzymała- Busse, Lech Polkowski and Andrzej Skowron

Fig. 1.43. On the way to Banff, RSCTC 2000; (from the right) Lech Polkowski, Jerzy Grzymała- Busse with wife Dobroslawa, and Professor Zdzisław Pawlak

Fig. 1.44. Visit in Kansas, Professor Zdzisław Pawlak on swing with (Witold and Jan) sons of Jerzy Grzymała-Busse

Fig. 1.45. Visit in Kansas, Professor Zdzisław Pawlak enjoying activities on swing in Kansas

Fig. 1.46. Lunch with Professor Zdzisław Pawlak in 2000 during the visit of Sankar K. Pal in Warsaw; (from the right) Sankar K. Pal, Professor Pawlak, Lech Polkowski, and Andrzej Skowron

Fig. 1.47. Rough Set Theory and Granular Computing (RSTGC 2001), Matsue, Shimane, Japan, May 20-22, 2001; (from the left) Shusaku Tsumoto with family, Sankar K. Pal, Professor Zdzisław Pawlak, Jerzy Grzymała-Busse, Paulina Zalewska and her friend

Fig. 1.48. Professor Zdzisław Pawlak at home with Andrzej Skowron (2000)

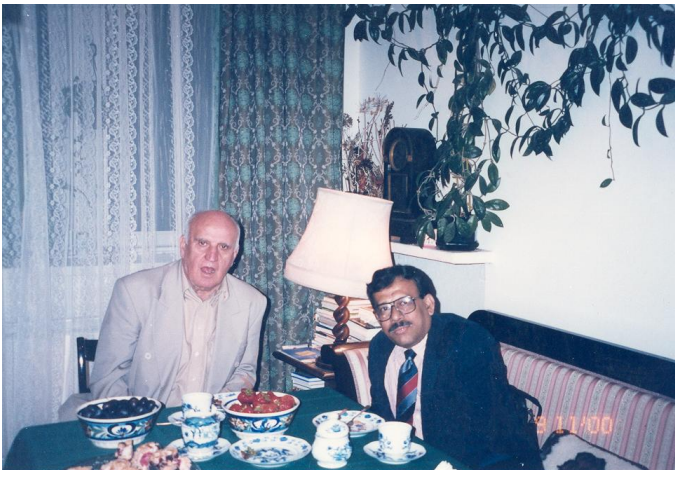


Fig. 1.49. Professor Zdzisław Pawlak singing arias at home (2000), (on the right) Sankar K. Pal

Fig. 1.50. Christmas 2003: Professor Zdzisław Pawlak and Andrzej Jankowski

Fig. 1.51. Christmas 2003: Professor Zdzisław Pawlak and Andrzej Skowron

Fig. 1.52. Professor Zdzisław Pawlak

1.9 Conclusions

Here we conclude our celebration of Professor Pawlak's life with few testimonies from students and collaborators. Other testimonies can be found in [6, 38, 39, 40, 48, 41]. A volume of Fundamenta Informaticae [46] includes articles dedicated to Professor Pawlak.
During one of the reminiscence sessions devoted to Professor Pawlak, Roman Słowiński stated:

> *The road which led Professor Pawlak to his crucial discoveries was long but ended successfully. Over fifty years, Professor Pawlak researched many areas of computer science. Without hesitation one can say that his personal path is one of the most important ones from the fifty-year-old history of research in Polish and worldwide computer science."*

During the same session, James Peters and Andrzej Skowron recalled the Professor:

> *Zdzisław Pawlak gave an abundance of his time and energy to help others. His personality and insight had, undoubtedly, influenced many scientists around the World. He had a unique gift to inspire his students, co-workers and many scientists beyond his close circle of collaborators to do research. Professor's associates recognized his extraordinary character. Many called him "Our Papa Pawlak".*
>
> *[...] Professor Zdzisław Pawlak was with us only for a while. However, when we consider his talents and great achievements, we know how much he influenced us and our successes with his research work in many areas like approximate reasoning, intelligent information*

systems, computation models and foundations of computer science, and of artificial intelligence - especially including rough set theory, molecular computing, pattern recognition, philosophy, art, and poetry. He also influenced us with his uncommonly rich personality.

Alicja Wakulicz-Deja farewelled Professor saying [51]:

Our meetings lasted to his last days.

[...] He seemed indestructible. However, his struggle with severe illness was very hard. When I was afraid that he was getting tired during our meetings and ask him if anything hurts him, he replied: "Let's not talk about it, others suffer more despite the fact they are better than me (like the Holy Father John Paul II)". Only at the end, there were moments, when he said: "You have better connections up there[9]*, tell them to take me now".*

[...] He often reiterated that scientific research is very hard and sometimes he would prefer to be a lumberjack, who may rest after the work, being surrounded by a beautiful nature. Nature often lured him to primeval forests and lakes. He documented it in his beautiful photographs and painted pictures.

[...] I think that he rests somewhere among his favourite forests and lakes now but I sometimes miss his words "I haven't seen you for some time, you are getting insubordinate recently Madame Professor".

We close this chapter with some pictures of our meetings with Professor Pawlak.

Acknowledgements. The authors would like to express gratitude to many Colleagues who supported in different ways the preparation of this chapter, in particular to Mohua Banerjee, Andrzej Czyżewski, Piotr Dembiński, Jerzy Fiett, Jerzy Dembczyński, Salvatore Greco, Anna Gomolińska, Wojciech Gomoliński, Andrzej Jankowski, Bożena Kostek, Solomon Marcus, Toshinori Munakata, Hung Son Nguyen, Hiroakira Ono, Lech Polkowski, Maria Semeniuk-Polkowska, Zbigniew W. Raś, Janusz Sosnowski, Urszula Stańczyk, Marcin Szczuka, Mirosław Truszczyński and Lotfi A. Zadeh.

At the beginning of this article is given only the corresponding author affiliation. Here, we include the affiliations of all other co-authors:

1. Mihir Kr. Chakraborty: Center for Cognitive Science, Jadavpur University, Kolkata 700032, India and Indian Statistical Institute, Kolkata, India, e-mail: mihirc4@gmail.com;
2. Jerzy Grzymała-Busse: Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA and Institute of Computer Science, Polish Academy of Sciences, 01–237 Warsaw, Poland, e-mail: jerzy@ku.edu;
3. Victor Marek: Department of Computer Science, University of Kentucky, Lexington, KY 40506, e-mail: marek@cs.uky.edu;
4. Sankar K. Pal: Indian Statistical Institute, 203 Barrackpore Trunk Road, Kolkata - 700 108, India, e-mail: sankar@isical.ac.in;
5. James F. Peters: Department of Electrical & Computer Engineering, University of Manitoba, 75A Chancellor's Circle, EITC Room E1-526, Winnipeg, Manitoba, R3T 3E2, Canada, e-mail: jfpeters@ee.umanitoba.ca;
6. Grzegorz Rozenberg: Leiden Center for Natural Computing, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands,e-mail: rozenber@liacs.nl;

[9] That is, in heaven (editor's note).

7. Dominik Ślęzak: Institute of Mathematics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland and Infobright Inc., Krzywickiego 34, lok. 219, 02-078 Warsaw, Poland, e-mail: slezak@{mimuw.edu.pl,infobright.com};
8. Roman Słowiński: Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland e-mail: rslowinski@cs.put.poznan.pl;
9. Alicja Wakulicz-Deja: Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland e-mail: wakulicz@us.edu.pl;
10. Shusaku Tsumoto: Department of Medical Informatics, Shimane University, School of Medicine, 89-1 Enya-cho, Izumo-city, Shimane 693-8501 Japan, e-mail: tsumoto@med.shimane-u.ac.jp;
11. Guoyin Wang: Chongqing University of Posts and Telecommunications, Chongqing, P. R. China 400065, e-mail: wanggy@ieee.org;
12. Wojciech Ziarko: Department of Computer Science, University of Regina, Regina, Saskatchewan, S4S 0A2 Canada, e-mail: ziarko@sasktel.net.

The research of Andrzej Skowron has been partially supported by the grant NN516 077837, from the Ministry of Science and Higher Education of the Republic of Poland, the grant 2011/01/D/ST6/06981 from the Polish National Science Centre, the National Centre for Research and Development (NCBiR) under grant SP/I/1/77065/10 by the strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information", and under the individual research grant by the program Homing Plus, edition 3/2011, from the Foundation for Polish Science.

References

1. Bar-Hillel, Y., Carnap, R.: An Outline of a Theory of Semantic Information. Research Laboratory of Electronics, Cambridge, MA, Technical Report 247. MIT (1953)
2. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press (1997)
3. Dembczyński, J.: Introduction. In: Długosz, K. (ed.) Zdzisław Pawlak. Doctor Honoris Causa of Poznań University of Technology, p. 5. Poznań University of Technology, Poznań (2002) (in Polish)
4. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques: A Rough Set Approach. STUDFUZZ, vol. 202. Springer, Heidelberg (2006)
5. Dretske, F.: Knowledge and the Flow of Information. The MIT Press, Cambridge (1981)
6. Ehrenfeucht, A., Peters, J.F., Rozenberg, G., Skowron, A.: Zdzisław Pawlak life and work (1926-2006). Bulletin of the European Association for Theoretical Computer Science (EATCS) 90, 202–204 (2006)
7. Floridi, L.: Information – A Very Short Introduction. Oxford University Press, Oxford (2010)
8. Floridi, L. (ed.): The Cambridge Handbook of Information and Computer Ethics. Cambridge University Press, Cambridge (2010)
9. Goldin, D., Smolka, S., Wegner, P.: Interactive Computation: The New Paradigm. Springer, Heidelberg (2006)
10. Head, T.: Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors. Bulletin of Mathematical Biology 49, 737–759 (1987)

11. Jankowski, A., Skowron, A.: Logic for artificial intelligence: The Rasiowa – Pawlak school perspective. In: Ehrenfeucht, A., Marek, V., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 106–143. IOS Press, Amsterdam (2008)
12. Joshi, M., Bhaumik, R.N., Lingras, P., Patil, N., Salgaonkar, A., Ślęzak, D.: Rough Set Year in India 2009. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 67–68. Springer, Heidelberg (2009)
13. Marcus, S.: Z. Pawlak a Precursor of DNA Computing and of Picture Grammars. In: Skowron, A. (ed.) New Frontiers in Scientific Discovery – Commemorating the Life and Work of Zdzisław Pawlak. Fundamenta Informaticae, vol. 75(1-4), pp. 331–334 (2007)
14. Marek, W., Pawlak, Z.: On the foundation of information retrieval. Bulletin of the Polish Academy of Sciences. Ser. Sci. Math. Astronom. Phys. 22(4), 447–452 (1974)
15. Marek, W., Pawlak, Z.: Information storage and retrieval - Mathematical foundations. Theoretical Computer Science 1, 331–354 (1976)
16. Munakata, T.: A Personal View on AI, Rough Set Theory and Professor Pawlak. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 247–252. Springer, Heidelberg (2007)
17. Pawlak, Z.: Flip-flop as generator of random binary digits. Mathematical Tables and Other Aids to Computation 10(53), 28–30 (1956)
18. Pawlak, Z.: Do they think, or not? Principles of operation of digital apparatuses. Problemy, 77–86 (1956) (in Polish)
19. Pawlak, Z.: Artificial intelligence. Argumenty 7, 4 (1963) (in Polish)
20. Pawlak, Z.: Organization of address-less computers working in parenthesis notation. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 9, 243–249 (1963)
21. Pawlak, Z.: Grammar and Mathematics. PZWS, Warsaw (1965) (in Polish)
22. Pawlak, Z.: Organization of Address-less Computers. PWN, Warszawa (1965) (in Polish)
23. Pawlak, Z.: On the notion of a computer. In: van Rootselaar, B., Staal, J.,F. (eds.) Logic, Methodology and Philosophy of Science III: Proceedings of the Third International Congress for Logic, Methodology and Philosophy of Science, pp. 255–267. North Holland, Amsterdam(1968)
24. Pawlak, Z.: Another comment on 'negative radix conversion'. IEEE Transactions on Computers (Correspondence) 20, 587 (1971)
25. Pawlak, Z.: Mathematical foundation of information retrieval. In: Proceedings of the International Symposium and Summer School on Mathematical Foundations of Computer Science, Strbske Pleso, High Tatras, Czechoslovakia, pp. 135–136. Mathematical Institute of the Slovak Academy of Sciences (1973)
26. Pawlak, Z.: About information retrieval. Delta 4, 12–14 (1975) (in Polish)
27. Pawlak, Z.: Information systems theoretical foundations. Information Systems 6(3) 205–218 (1981)
28. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
29. Pawlak, Z.: On conflicts. International Journal of Man-Machine Studies 21, 127–134 (1984)
30. Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning about Data. System Theory, Knowledge Engineering and Problem Solving 9 (1991)
31. Pawlak, Z.: Anatomy of conflics. Bulletin of the EATCS 50, 234–246 (1993)
32. Pawlak, Z.: Thanks. In: Długosz, K. (ed.) Zdzisław Pawlak. Doktor Honoris Causa of Poznań University of Technology, pp. 15–18. Poznań University of Technology, Poznań (2002) (in Polish)

33. Pawlak, Z.: What is next? In: Długosz, K. (ed.) Zdzisław Pawlak. Doctor Honoris Causa of Poznań University of Technology, pp. 21–29. Poznań University of Technology, Poznań (2002) (in Polish)
34. Pawlak, Z.: Some remarks on conflict analysis. European Journal of Operational Research 166(3), 649–654 (2005)
35. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
36. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177(1), 28–40 (2007)
37. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177(1), 41–73 (2007)
38. Peters, J.F., Skowron, A.: Some Contributions by Zdzisław Pawlak. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 1–11. Springer, Heidelberg (2006)
39. Peters, J.F., Skowron, A.: Zdzisław pawlak: Life and work. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 1–24. Springer, Heidelberg (2006)
40. Skowron, A., Peters, J.F.: Zdzisław Pawlak Commemorating His Life and Work. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 49–52. Springer, Heidelberg (2006)
41. Peters, J.F., Skowron, A.: Zdzisław Pawlak life and work (1926-2006). Information Sciences 171 (1), 1–2 (2007)
42. Polkowski, L., Skowron, A. (eds.): RSCTC 1998. LNCS (LNAI), vol. 1424. Springer, Heidelberg (1998)
43. Rosenfeld, A.: Picture Processing by Computer. Academic Press, New York – London (1969)
44. Salomaa, A.: Formal Languages. Academic Press, New York – London (1973)
45. Shaw, A.C.: A proposed language for the formal description of pictures. GSG Memo 28, Stanford, California, pp. 1–32 (February 1967)
46. Skowron, A.: New frontiers in scientific discovery – Commemorating the life and work of Zdzisław Pawlak. Fundamenta Informaticae 75(1-4), 1–562 (2007)
47. Słowiński, R.: Laudatio dedicated to Mr Professor Ph.D. hab. M.Eng. Zdzisław I. Pawlak. In: Długosz, K. (ed.) Zdzisław Pawlak. Doctor Honoris Causa of Poznań University of Technology, pp. 7–11. Poznań University of Technology, Poznań (2002) (in Polish)
48. Słowiński, R.: Prof. Zdzisław Pawlak (1926-2006). Obituary. Fuzzy Sets and Systems 157, 2419–2422 (2006)
49. Stańczyk, U. (ed.): Adam Mrózek (1948-1999). Institute of Theoretical and Applied Informatics of PAS, p. 13, Gliwice (2009) (in Polish)
50. Vitória, A., Maluszynski, J., Szałas, A.: Modeling and reasoning with paraconsistent rough sets. Fundamenta Informaticae 97(4), 405–438 (2009)
51. Wakulicz-Deja, A. (ed.): Decision Support Systems volume I. University of Silesia, Katowice, 1–206 (2007) (in Polish)
52. Wang, G., Zhang, Q., Huang, H.-K., Ye, D., Hu, Q.-H., Hu, X.-G., Shi, Z.-Z., Li, Y., Shang, L., An, L., Sai, Y., Chen, S., Liang, J., Qin, K., Zeng, H., Xie, K., Miao, D., Min, F., Wu, Z., Wu, W.-Z., Dai, J.-H.: Research on Rough Set Theory and Applications in China. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets VIII. LNCS, vol. 5084, pp. 352–395. Springer, Heidelberg (2008)

53. Wang, G., Zhang, Q., Yu, H., Hu, J., Hu, F.: Expressing and processing of uncertain knowledge. In: International Workshop on Rough Sets: Future Research Directions, June 26-27. Faculty of Mathematics, Computer Science and Mechanics at Warsaw University, Warsaw (2010)
54. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29, 81–95 (1988); reprinted in Gains, B., Boose, J. (eds), Machine Learning and Uncertain Reasoning, vol. 3, pp. 227–242. Academic Press (1990)
55. http://www.cs.utep.edu/interval-comp/pawlak.html
56. Rough Set Database System, http://rsds.univ.rzeszow.pl/

Chapter 2
List of Works by Professor Zdzisław Pawlak (1926-2006)

Andrzej Skowron

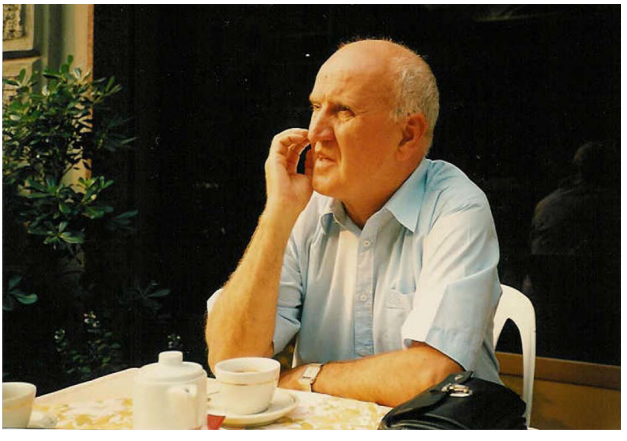


Fig. 2.1. Professor Zdzisław Pawlak, 2004

Publications of Professor Zdzisław Pawlak

1953–1960

1. O rachunku prawdopodobieństwa (On probability calculus). Problemy, 203–204 (1953) (in Polish)
2. Flip-Flop as generator of random binary digits. Mathematical Tables and Other Aids to Computation **10**(53), 28–30 (1956)
3. Myślą czy nie myślą? (Are they thinking or not?) Problemy, 77–86 (1956) (in Polish)
4. (with A.Wakulicz) Use of expansion with a negative base in the arithmometer of a digital computer. Bull. Acad. Pol. Sci., Sér. Sci. Math. Astronom. Phys. **5**(3) 233–236 (1957)

Andrzej Skowron
Institute of Mathematics
The University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
e-mail: skowron@mimuw.edu.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 57–74.
springerlink.com

5. Decoding Nets and the Theory of Graphs. Journal of the Society for Industrial and Applied Mathematics 7(1) (Mar., 1959) 1–5 (published by: Society for Industrial and Applied Mathematics)
6. An Electronic Computer Based on the "-2" System. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **7** 713–721 (1959)

1960–1969

7. Organization of digital computers and computable functions. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(1), 41–51 (1960)
8. The Application of negative base number system to differential analyzer. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(3), 149–157 (1960)
9. The Application of systematic binary expansions to decimal codes. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(3), 151–152 (1960)
10. Organization of address-free digital computer for calculation of simple arithmetic expressions. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(4), 193–195 (1960)
11. New method of parenthesis-free notation of formulas. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(4), 197–198 (1960)
12. The organization of digital computer based on "-2" System. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(5), 253–258 (1960)
13. Automatic programming of arithmetic formulae in parenthesis notation by the addressing function. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(6), 315–319 (1960)
14. On the application of the rule of substitution in the organization of an address-free computer. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(11-12), 681–684 (1960)
15. On the realization of recursive schemes in the address-free computer. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **8**(11-12), 685–687 (1960)
16. Organization of address-free computer with separate memory for partial results. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(2), 123–127 (1961)
17. Some remarks on "-2" digital computer. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(4), 22–28 (1961)
18. Organization of address-free computer B-100. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(4), 229–234 (1961)
19. New conception of two address computer. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(5), 313–315 (1961)
20. Some remarks on automatic programming of arithmetic formulae. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(5), 317–320 (1961)
21. Realization of memory of partial results in certain parenthesis free formalism. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(8) 487–495 (1961)
22. On the utility of arithmetical formalism in digital computers. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(9), 527–529 (1961)
23. Realization of the rule of substitution in addressless computers. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(9), 531–534 (1961)
24. Elimination of waiting time in serial address-less computers. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(10), 577–588 (1961)
25. Realization of rule of substitution in the addressless computer without working memory. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(10), 579–580 (1961)
26. The application of two-valued lattices to the realization of many-valued truth-tables. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(11), 655–658 (1961)
27. Realization of certain classes of recursive formulae in the addressless computer. Bull. Acad. Pol. Sci., Sér. Sci. Tech. **9**(11), 651–653 (1961)

28. (with R. Bartoszyński) Matematyczne przyczynki do rozwoju prakseologii (Mathematical contribution to praxeology development). Warszawa 1963. http://pl.wikipedia.org/wiki/Klemens_Szaniawski (in Polish)
29. New class of mathematical languages and organization of address-less computers. In: Proc. Colloquium on the Foundation of Mathematics, Tihany 1962, pp. 227–243. Budapest (1965)
30. Organization of address-less computers working in parenthesis notation. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **9**, 243–249 (1963)
31. Maszyny i matematyka (Machines and mathematics). Problemy, 313–314 (1963) (in Polish)
32. Mechaniczne tłumaczenie (Automatic translation). Problemy, 546–547 (1963) (in Polish)
33. Sztuczna inteligencja (Artificial Intelligence). Argumenty (7) 4 (1963) (in Polish)
34. Całość i rozwój w świetle cybernetyki. Dyskusja o książce O. Langego (Whole and development from the point of view of cybernetics. Discussion on the book by O. Lange). Studia Filozoficzne (3-4), 27–28 (1963) (in Polish)
35. Maszyna i język (Machine and Language). Warszawa, PWN, Warszawa (1964) (in Polish)
36. Organizacja maszyn bezadresowych (Organization od Address-less Computers). PWN, Warszawa (1965) (in Polish)
37. Gramatyka i matematyka (Grammar and Mathematics). PZWS, Warszawa (1965) (in Polish)
38. Automatyczne dowodzenie twierdzeń (Automated Theorem Proving). PZWS, Warszawa (1965) (in Polish)
39. Sygnały, symbole, maszyny (Signals, Symbols, Machines). Nowości Nauki i Techniki. Wiedza Powszechna, Warszawa, 1966 (in Polish)
40. Matematyczna teoria procesu składania (Mathematical Theory of Assembly Process). Prekseologia 23, 177–202 (1966) (in Polish)
41. (with A. Ehrenfeucht) Some remarks on parenthesis-free languages. Bull. Acad. Pol. Sci., Sér. Sci. Math. Astronom. Phys. **15**(2), 105–106 (1967)
42. On the notion of a computer. In: B. van Rootselaar, J. F. Staal (eds.) Logic, Methodology and Philosophy of Science III: Proceedings of the Third International Congress for Logic, Methodology and Philosophy of Science, Amsterdam 1967, pp. 255–267. North Holland, Amsterdam (1968)
43. Matematyczne aspekty procesu produkcyjnego (Mathematical Aspects of Production Process). Polskie Wydawnictwo Ekonomiczne, Warszawa (1969) (in Polish)
44. Uwagi o teorii maszyn cyfrowych (Remarks on theory of computers). Maszyny Matematyczne **10**, 4–7 (1969)(in Polish)
45. Maszyny matematyczne (Mathematical machines). Roczniki Polskiego Towarzystwa Matematycznego, Ser. 2, 12, 109–114 (1969) (in Polish)
46. Maszyny programowane (Stored program machines). Algorytmy **5**(10) 5–19 (1969) (in Polish)

1970–1979

47. Podstawy matematyczne maszyn cyfrowych (Foundations of digital machines). In: Naukowe problemy maszyn matematycznych. Materiały z I Ogólnopolskiego Sympozjum: Naukowe Problemy Maszyn Matematycznych, 21-26 października 1968, pp. 71–77. PWN, Waszawa (1970) (in Polish)

48. Definitional Approach to Automatic Demonstration. In: M. Laudet, D. Lacombe, L. Nolin, and M. Schützenberger (eds.) Proceedings of the Symposium on Automatic Demonstration, Versailles, France, December 1968, *Lecture Notes in Mathematics*, vol. 125, pp. 191–193. Springer-Verlag, Berlin (1970)
49. (with A. W. Mostowski) Logika dla inżynierów. (Logic for Engineers) PWN, Warszawa (1970) (in Polish)
50. About the meaning of personal pronouns. *CCPAS Reports*, vol. 16, pp.1–7. Computational Centre Polish Academy of Sciences, Warsaw (1970)
51. Maszyny Matematyczne. (Mathematical machines) PZWS, Warszawa (1971) (in Polish)
52. Another comment on 'Negative radix conversion'. IEEE Transactions on Computers (Correspondence) **20**(5), 587 (May, 1971)
53. Mathematical foundations of computers. In: Proceedings of the International Symposium and Summer School on Mathematical Foundations of Computer Science, August 21-27, 1972, Warsaw, Jabłonna 1972, pp. 1–12. Computation Center of the Polish Academy of Sciences and Institute of Computing Machines of Warsaw University, Warsaw (1972)
54. Zastosowania maszyn matematycznych a zastosowania matematyki (Applications of mathematical machines and applications of mathematics). Roczniki Polskiego Towarzystwa Matematycznego, Ser. 2, 14, 13–16 (1972) (in Polish)
55. Mathematical foundations of information retrieval. *CCPAS Reports*, vol. 101, pp. 1–8. Computation Centre, Polish Academy of Sciences, Warsaw (1973)
56. (with W. Marek) Mathematical foundations of information storage and retrieval. Part 1. *CC PAS Reports*, vol. 135, pp. 1–10. Computation Centre, Polish Academy of Sciences, Warsaw (1973)
57. (with W. Marek) Mathematical foundations of information storage and retrieval. Part 2. *CC PAS Reports*, vol. 136, pp. 1–11. Computation Centre, Polish Academy of Sciences, Warsaw (1973)
58. (with W. Marek) Mathematical foundations of information storage and retrieval. Part 3. *CC PAS Reports*, vol. 137, pp. 1–8. Computation Centre, Polish Academy of Sciences, Warsaw (1973)
59. About the meaning of personal pronouns. Cahiers de Linguistique Theoretique and Appliquee **3**, 261–269 (1973)
60. A mathematical model of digital computer. In: Proceedings of the Conference on Automata and Formal Languages, Bonn, 1973, *Lecture Notes in Computer Science*, vol. 2, pp. 16–22. Springer-Verlag, Berlin (1973)
61. Mathematical foundation of information retrieval. In: Proceedings of the International Symposium and Summer School on Mathematical Foundations of Computer Science, Strbske Pleso, High Tatras, Czechoslovakia, 1973. pp. 135–136. Mathematical Institute of the Slovak Academy of Sciences (1973)
62. (with W. Marek) Information storage and retrieval system - Mathematical foundations. *CC PAS Reports*, vol. 149, pp. 1–49. Computation Centre, Polish Academy of Sciences, Warsaw (1974)
63. (with W. Marek) On the foundation of information retrieval. Bull. Pol. Acad. Sci. Math. **22**(4), 447–452 (1974)
64. Rozważania o informatyce (Considerations on informatics). Problemy, 2–6 (1974) (in Polish)
65. O wyszukiwaniu informacji (On information retrieval). Delta **4**, 12–14 (1975) (in Polish)

66. Teoretyczne problemy systemów wyszukiwania informacji (Theoretical problems of information retrieval systems). Aktualne Problemy Informacji i Dokumentacji **5**, 47–50 (1975) (in Polish)
67. (with W. Marek) Information storage and retrieval mathematical foundations. Theoretical Computer Science **1**, 331–354 (1976)
68. (with A. W. Mostowski) Logica para Enginieros. Escuela de Filosofia, Universidad del Zulia, Venezuela, Maracaibo (1976) (translation of [49])
69. (with A. Mazurkiewicz (eds.)) Mathematical Foundations of Computer Science, *Banach Center Publications* **2**, PWN, Warsaw, Poland (1977)
70. Computers and programs. Computer Education **13**, 17–19 (1978)
71. Information systems. *ICS PAS Reports*, vol. 338, pp. 1–15. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1978)
72. (ed.) Problemy sterowania w sieciach komputerowych (Control Problems in Computer Networks). Sprawozdania z Seminarium Polsko-Rysyjskiego, Sierpień 25-28, 1978, Warszawa, PWN, Warszawa - Łódź 1979 (in Russian)
73. (with M. Muraszkiewicz, H. Rybiński) Elementy teorii systemów informacyjnych (Elements of Information Systems Theory). Instytut Informacji Naukowej, Technicznej i Ekonomicznej, Warszawa 1985 (in Polish)
74. Distributed information systems. *ICS PAS Reports*, vol. 370, pp. 1–28. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1979)
75. (with G. Rozenberg and W. J. Savitch) Programs for instruction machines. Information and Control **41** 9–28 (1979)

1980–1989

76. Toward the theory of information systems. The notion of information system. *ICS PAS Reports*, vol. 419, 1–35. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1980)
77. Classification of objects by means of attributes. *ICS PAS Reports*, vol. 429, pp. 1–20. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
78. Rough sets. Basic notions. *ICS PAS Reports*, vol. 431, pp. 1–12. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
79. (with E. Orłowska) Expressive power of knowledge representation systems. *ICS PAS Reports*, vol. 432, pp. 1–31. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
80. (with E. Konrad, E. Orłowska) Knowledge representation systems. Definability of informations. *ICS PAS Reports*, vol. 433, pp. 1–44. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
81. Rough relations. Institute of Computer Science, *ICS PAS Reports*, vol. 435, pp. 1–10. Polish Academy of Sciences, Warsaw (1981)
82. (with W. Marek) Rough sets and information systems. *ICS PAS Reports*, vol. 441, pp. 1–15. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
83. (with E. Orłowska) Representation of nondeterministic information. *ICS PAS Reports*, vol. 450, pp. 1–24. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
84. About conflicts. *ICS PAS Reports*, vol. 451, pp. 1–20. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1981)
85. Information systems theoretical foundations. Information Systems **6**(3) 205–218 (1981)

86. (with R. Brooks, M. Blattner, E. Barrett) Using partitioned databases for statistical data analysis. In: 1981 American Federation of Information Processing Societies (AFIPS) National Computer Conference, Chicago, IL, May 4-7, 1981, AFIPS Conference Proceedings 50, pp. 453–457. AFIPS Press, Chicago, IL (1981)
87. (with E. Konrad, and E. Orłowska) On approximate concept learning. In: Proceedings of the European Conference on Artificial Intelligence **11**(5), Orsay, France 1982, pp. 17–19. IOS Press, Amsterdam (1982) (see also Technische Universität Berlin, Bericht Nr. 81-7, October 1981, 1–21).
88. Some remarks about rough sets. *ICS PAS Reports*, vol. 456, pp. 1–11. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1982)
89. Rough functions. *ICS PAS Reports*, vol. 467, pp. 1–11. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1982)
90. Rough sets. Power set hierarchy. *ICS PAS Reports*, vol. 470, pp. 1–27. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1982)
91. Rough sets. Algebraic and topological approach. *ICS PAS Reports*, vol. 482, pp. 1–32. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1982)
92. Rough probability and partial observability. *ICS PAS Reports*, vol. 496, pp. 1–20. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1982)
93. Rough sets. International Journal of Computer and Information Sciences **11** 341–356 (1982)
94. Rough Classification. *ICS PAS Reports*, vol. 506, pp. 1–31. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1983)
95. Discrimination power of attributes in knowledge representation system. *ICS PAS Reports*, vol. 553, pp. 1–15. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1983)
96. Systemy informacyjne. Podstawy teoretyczne (Information Systems. Theoretical Foundations). WNT, Warszawa 1983 (in Polish)
97. (with M. Novotny) On a representation of rough sets by means of information systems. Fundamenta Informaticae **6**(3-4) 289–296 (1983)
98. (with E. Orłowska) Logical foundations of knowledge representation. *ICS PAS Reports*, vol. 537, pp. 1–108. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1984)
99. Rough sets and fuzzy sets. *ICS PAS Reports*, vol. 540, pp. 1–10. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1984)
100. Rough classification. Journal of Man-Machine Studies **20**, 469–483 (1984)
101. On conflicts. Int. J. of Man-Machine Studies **21**, 127–134 (1984)
102. On discernibility of objects in knowledge representation systems. Bull. Pol. Acad. Sci., Math. **32**(9-10), 613–615 (1984)
103. Rough probability. Bull. Pol. Acad. Sci., Math. **32**(9-10), 607–612 (1984)
104. On superfluous attributes in knowledge representation system. Bull. Pol. Acad. Sci., Tech. Sci. **32**(3-4), 211–213 (1984)
105. Principles of knowledge representation. Bull. of the Section of Logic **12**(4), 194–201 (1984)
106. On rough sets. Bull. of the European Association for Theoretical Computer Science **24**, 94–109 (1984)
107. (with J. W. Grzymała-Busse) On some subsets of the partition set. Fundamenta Informaticae **7**(4), 483–488 (1984)
108. (with W. Marek) Rough sets and information systems. Fundamenta Informaticae **7**(1), 105–115 (1984)

109. (with E. Orłowska) Measurement and indiscernibility. Bull. Pol. Acad. Sci., Math. **32**(9-10), 617–624 (1984)
110. (with E. Orłowska) Representation of nondeterministic information. Theoretical Computer Science **29**, 27–39 (1984)
111. (with E. Orłowska) Expressive power of knowledge representation. International Journal of Man Machine Studies **20**, 458–500 (1984)
112. Rough sets and fuzzy sets. *ICS PAS Reports*, vol. 540, pp. 1–7. Institute of Computer Science, Polish Academy of Sciences Warsaw (1984)
113. Rough sets and some problems of artificial intelligence. *ICS PAS Reports*, vol. 565, pp. 1–55. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1985)
114. Rough sets and fuzzy sets. Journal of Fuzzy Sets and Systems **17**, 99–102 (1985)
115. Decision tables and decision algorithms. Bull. Pol. Acad. Sci., Tech. Sci. **33**(9-10), 487–494 (1985)
116. Some Remarks on rough sets. Bull. Pol. Acad. Sci., Tech. Sci. **3**(11-12), 567–572 (1985)
117. Rough concept analysis. Bull. Pol. Acad. Sci., Tech. Sci. **33**(9-10) 495–498 (1985)
118. Rough sets and multi expert systems. Bull. Pol. Acad. Sci., Tech. Sci. **33**(9-10) 499–504 (1985)
119. On rough dependency of attributes in information systems. Bull. Pol. Acad. Sci., Tech. Sci. **33**(9-10), 481–485 (1985)
120. (with W. Marek) One-dimensional learning. Fundamenta Informaticae **8**(1), 83–88 (1985)
121. (with M. Novotny) Characterization of rough top equalities and rough bottom equalities. Bull. Pol. Acad. Sci., Math. **33**(1-2), 91–97 (1985)
122. (with M. Novotny) On rough equalities. Bull. Pol. Acad. Sci., Math. **33**(1-2), 99–104 (1985)
123. (with M. Novotny) Black box analysis and rough top equalities. Bull. Pol. Acad. Sci., Math. **33**(1-2), 105–113 (1985)
124. (with C. Rauszer) Dependency of Attributes in Information Systems. Bull. Pol. Acad. Sci., Math. **33**, 551–559 (1985)
125. Rough sets and decision tables. In: A. Skowron (ed.) Proceedings of the 5th Symposium on Computation Theory, Zaborów, Poland December 3-8, 1984, *Lecture Notes in Computer Science*, vol. 208, pp. 186–196. Springer-Verlag, Heidelberg (1986)
126. On learning - A rough set approach. In: A. Skowron (ed.) Proceedings of the 5th Symposium on Computation Theory, Zaborów, Poland December 3-8, 1984, Lecture Notes in Computer Science 208, Springer-Verlag, Heidelberg (1986) 197–227
127. On decision tables. Bull. Pol. Acad. Sci., Tech. Sci. **34**(9-10), 563–566 (1986)
128. Learning from examples. Bull. Pol. Acad. Sci., Tech. Sci. **34**(9-10), 573–586 (1986)
129. On rough relations. Bull. Pol. Acad. Sci., Tech. Sci. **34**(9-10), 587–590 (1986)
130. Decision table computer. Bull. Pol. Acad. Sci., Tech. Sci. **34**(9-10) 591–595 (1986)
131. Rough sets. Fuzzy Sets and Systems **19**, 309–310 (1986)
132. (with K. Słowiński and R. Słowiński) Rough classification of patients after highly selected vagotomy for duodenal ulcer. Journal of Man-Machine Studies **24**, 413–433 (1986)
133. O konfliktach (On Conflicts). PWN, Warszawa 1987 (in Polish)
134. On rough functions. Bull. Pol. Acad. Sci., Tech. Sci. **35**(5-6), 249–251 (1987)
135. Rough Logic. Bull. Pol. Acad. Sci., Tech. Sci. **35**(5-6), 253–258 (1987)
136. Learning from examples the case of an imperfect teacher. Bull. Pol. Acad. Sci., Tech. Sci. **35**(5-6), 259–264 (1987)

137. Decision tables a rough set approach. Bulletin of the European Association for Theoretical Computer Science (EATCS) **33**, 85–96 (1987)
138. (with J. Fibak, R. Słowiński and K. Słowiński). Rough sets based decision algorithm for treatment of duodenal ulcer by HSV. Bull. Polish Acad. Sci. Biol. **34**(9-10), 227–246 (1987)
139. (with M. Novotny) Concept forming and black boxes. Bull. Pol. Acad. Sci., Math. **35**(1-2), 133–141 (1987)
140. Approximate classification and rough sets. In: R. Wille (ed.), Klassifikation und Ordnung, pp. 186–190. Tagungsband, Frankfurt/Main (1988)
141. Communication logic. Bull. Pol. Acad. Sci., Tech. Sci. **36**(1-2), 107–111 (1988)
142. Hard sets and soft sets. Bull. Pol. Acad. Sci., Tech. Sci. **36**(1-2), 119–123 (1988)
143. Rough sets and information systems. Podstawy Sterowania **18**(3-4) 175–200 (1988)
144. On data reduction and analysis. A rough set approach. In: Proceedins of the FAW-Workshop "Mathematics and AI" vol. II, Schloss Reisensburg, December 19-22, FAW-Bericht, vol. B-89008, pp. 545–572. FAW Ulm, (1988)
145. (with M. Novotny) Partial dependency of attributes. Bull. Pol. Acad. Sci., Math. **36**(7-8), 453–458 (1988)
146. (with M. Novotny) Independence of attributes. Bull. Pol. Acad. Sci., Math. **36**(7-8), 459–465 (1988)
147. Knowledge, reasoning and classification. A rough set perspective. Bulletin of the European Association for Theoretical Computer Science **38**, 199–210 (1989)
148. Indiscernibility, partitions and rough sets. In: R. Narasimhan (ed.), Commemorative Volume on Theoretical Computer Science - In Honor of Prof.G. Siromony, pp. 210–220. World Scientific Publishing Comp. Co., Singapore (1989)

1990–1999

149. (with M. Novotny) On superreducts. Bull. Pol. Acad. Sci., Tech. Sci. **38**(1-2), 101–112 (1990)
150. (with S.K.M. Wong and W. Ziarko). Rough sets: Probabilistic versus deterministic approach. Journal of Man-Machine Studies **29**, 81–85 (1988) ; see also: B. Gains, J. Boose (eds.), Machine Learning and Uncertain Reasoning vol. 3, pp. 227–242. Academic Press, (1990)
151. Information systems and decision tables a rough set perspective. Archiwum Informatyki Teoretycznej i Stosowanej **2**(3-4), 139–166 (1990)
152. Decision logic. Bulletin of the European Association for Theoretical Computer Science (EATCS) **44**, 201–225 (1991)
153. Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston, Dordrecht (1991)
154. Rough Sets and their applications. *ICS Research Report*, vol. 18, pp. 1–9. Institute of Computer Science, Warsaw University of Technology (1992)
155. Wiedza z perspektywy zbiorów przybliżonych (Knowledge from perspective of rough sets). *ICS Research Report*, vol. 23, pp. 1–22. Institute of Computer Science, Warsaw University of Technology (1992) (in Polish)
156. Preface. In: R. Słowiński (ed.) Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, *System Theory, Knowledge Engineering and Problem Solving*. vol. 11, pp. ix–xi. Kluwer Academic Publishers, Dordrecht, The Netherlands (1992)
157. (with M. Novotny) On problem concerning dependence spaces. Fundamenta Informaticae **16**(3-4), 275–287, (1992)

158. Concurrent versus sequential - The rough set perspective. Bulletin of the European Association for Theoretical Computer Science (EATCS) **48**, 178–190 (1992)
159. Rough sets: A new approach to vagueness. In: L. Zadeh and J. Kacprzyk (eds.), Fuzzy Logic for the Management of Uncertainty, pp. 105–118. John Wiley & Sons, Inc., New York (1992)
160. Rough sets present state and the future. *ICS Research Report*, vol. 20, pp. 1–11. Institute of Computer Science, Warsaw University of Technology (1993)
161. (with R. Słowiński) Decision analysis using rough sets. *ICS Research Report*, vol. 21, pp. 1–18. Institute of Computer Science, Warsaw University of Technology (1993)
162. (with R. Słowiński) Rough set approach to multi-attribute decision analysis. *ICS Research Report*, vol. 36, pp. 1–31. Institute of Computer Science, Warsaw University of Technology (1993)
163. On some issues connected with conflict analysis. *ICS Research Report*, vol. 37, pp. 1–11. Institute of Computer Science, Warsaw University of Technology (1993)
164. Rough sets - Basic notions. *ICS Research Report*, vol. 52, pp. 1–15. Institute of Computer Science, Warsaw University of Technology (1993)
165. Rough sets. Present state and the future. In: R. Słowiński and J. Stefanowski (eds.) Proceedings of the First International Workshop on Rough Sets: State of the Art and Perspectives. Kiekrz - Poznań, Poland September 2-4 (1992), pp. 51–53 and Foundations of Computing and Decision Sciences *18*(3-4), 157–166 (1993)
166. Anatomy of conflict, Bulletin of the European Association for Theoretical Computer Science (EATCS) **50**, 234–247 (1993)
167. On some issues connected with conflict analysis. In: Proceedings of the Workshop on Intelligent Information Systems, Augustów, Poland, June 7-11, 1993, pp. 3–14. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1993)
168. (with A. Skowron) Rough membership functions: A tool for reasoning with uncertainty. In: C. Rauszer (ed.), Algebraic Methods in Logic and Computer Science, *Banach Center Publications*, vol. 28, pp. 135–150. Polish Academy of Sciences, Warsaw (1993)
169. (with A. Skowron) A rough set approach for decision rules generation. In: Proceedings of the Workshop W12: The Management of Uncertainty in AI at 13th IJCAI, Chambery Savoie, France, August 30, (1993) see also: Z. Pawlak, A. Skowron: A rough set approach for decision rules generation. *ICS Research Report*, vol. 23, pp. 1–19. Institute of Computer Science, Warsaw University of Technology (1993)
170. (with E. Czogała, A. Mrózek) The idea of a rough fuzzy controller and its applications to the stabilization of a pendulum-car system. *ICS Research Report*, vol. 7, pp. 1–23. Institute of Computer Science, Warsaw University of Technology (1994)
171. Hard and soft sets. *ICS Research Report*, vol. 10, pp. 1–6. Institute of Computer Science, Warsaw University of Technology (1994)
172. Vagueness and uncertainty: A rough set perspective. *ICS Research Report*, vol. 19, pp. 1–6. Institute of Computer Science, Warsaw University of Technology (1994)
173. Rough sets, rough relation, and rough functions. *ICS Research Report*, vol. 24, pp. 1–7. Institute of Computer Science, Warsaw University of Technology (1994)
174. An inquiry into vaguenes and uncertainty. *ICS Research Report*, vol. 29, pp. 1–7. Institute of Computer Science, Warsaw University of Technology (1994)
175. (with E. Czogała, A. Mrózek) Rough fuzzy controller as an approximation of fuzzy controller. *ICS Research Report*, vol. 32, pp. 1–7. Institute of Computer Science, Warsaw University of Technology (1994)
176. Rough sets present state and further prospects. *ICS Research Report*, vol. 49, pp..1–5. Institute of Computer Science, Warsaw University of Technology (1994)

177. Rough real functions. *ICS Research Report*, vol. 50, pp. 1–5. Institute of Computer Science, Warsaw University of Technology (1994)
178. Hard and soft sets. In: W. Ziarko (ed.), Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93), Workshops in Computing, Springer-Verlag & British Computer Society, pp. 130–135. London, Berlin (1994)
179. Knowledge and uncertainty A rough sets approach. In: V. Alagar, S. Bergler, F.Q. Dong (eds.) Incompleteness and Uncertainty in Information Systems, Proceedings of SOFTEKS Workshop on Incompleteness and Uncertainty in Information Systems, Concordia University, Montreal, Canada 1993, *Workshops in Computing*, pp. 34–42. Springer-Verlag & British Computer Society, London, Berlin (1994)
180. An inquiry into vagueness and uncertainty. In: M. Dąbrowski, M. Michalewicz, and Z.W. Raś (eds.), Proceedings of the Third International Workshop on Intelligent Information Systems, Wigry, Poland, June 6-10, 1994, pp. 338–343. Institute of Computer Science, Polish Academy of Sciences, Warsaw (1994)
181. Rough sets present state and further prospects. In: T.Y. Lin and A.M. Wildberger (eds.), Third International Workshop on Rough Sets and Soft Computing (RSSC94), November 10-12, 1994, San Jose, California, pp. 72–77. San Jose University, CA (1994)
182. Rough sets and their applications. Microcomputer Applications **13**(2) 71–75 (1994)
183. (with A. Skowron) Rough membership functions. In: R. Yaeger, M. Fedrizzi, and J. Kacprzyk (eds.), Advances in the Dempster Shafer Theory of Evidence, pp. 251–271. John Wiley & Sons, Inc., New York (1994)
184. (with R. Słowiński) Decision analysis using rough sets. International Transactions in Operational Research **1**(1), 107–114 (1994) (see also: Decision analysis using rough sets, Invited paper for XIII World Conference on Operation Research IFORS 93, Lisbon, Portugal July 12-16, 1993, and *ICS Research Report*, vol. 21, pp. 1–18. Institute of Computer Science, Warsaw University of Technology (1993))
185. (with R. Słowiński) Rough set approach to multi-attribute decision analysis. European Journal of Operational Research **72**, 443–459 (1994)
186. (with E. Czogała and A. Mrózek) Application of a rough fuzzy controller to the stabilization of an inverted pendulum. In: Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing Vol. 3, pp. 1403–1406. Aachen, Germany (1994)
187. (with K. Frąckowiak) Synthesis of the survey results. In: E. Klein, D. Griesbach (eds.) Survey of Language Engineering Organisations in Central and Eastern Europe, pp. 26–27. European Commission (1994)
188. Rough real functions and rough controllers. *ICS Research Report*, vol. 1, pp. 1–12. Institute of Computer Science, Warsaw University of Technology (1995)
189. Rough set approach to knowledge-based decision support. *ICS Research Report*, vol. 10, pp. 1–12. Institute of Computer Science, Warsaw University of Technology (1995)
190. On some issues connected with roughly continuous functions. *ICS Research Report*, vol. 21, pp. 1–6. Institute of Computer Science, Warsaw University of Technology (1995)
191. Zbiory przybliżone (Rough sets). *ICS Research Report*, vol. 31, pp. 1–8. Institute of Computer Science, Warsaw University of Technology (1995) (in Poilsh)
192. Rough sets present state and further prospects. *ICS Research Report*, vol. 32, pp. 1–14. Institute of Computer Science, Warsaw University of Technology (1995)
193. On rough derivatives, rough integrals and rough differential equations. *ICS Research Report*, vol.41, pp. 1–13. Institute of Computer Science, Warsaw University of Technology (1995)

194. Rough calculus. *ICS Research Report*, vol. 58, pp. 1–11. Institute of Computer Science, Warsaw University of Technology (1995)
195. (with E. Czogała and A. Mrózek) The idea of a rough fuzzy controller and its applications to the stabilization of a pendulum-car system. Fuzzy Sets and Systems **72**, 61–73 (1995)
196. On some issues connected with indiscernibility. In: G. Paun (ed.) Mathematical Linguistics and Related Topics, pp. 279–283. Editura Academiei Romane, Bucaresti (1995)
197. Rough sets. In: C.J. Hwang, B.W. Hwang (eds.) Proceedings of the 23rd ACM Annual Computer Science Conference, February 28 - March 2, 1995, Nashville, Tennessee, pp. 262–264. ACM, New York (1995)
198. Rough real functions and rough controllers. In: T.Y. Lin (ed.), Proceedings of the Workshop on Rough Sets and Data Mining at the 23rd ACM Annual Computer Science Conference, March 2, 1995, Nashville, Tennessee, pp. 58–64. ACM, New York (1995)
199. Vagueness and uncertainty: A rough set perspective. Computational Intelligence: An International Journal **11**(2) 227–232 (1995) (special issue edited by W. Ziarko)
200. Rough calculus. In: Proceedings of the Second Annual Joint Conference on Information Sciences, September 28 - October 1, Wrightsville Beach, North Carolina, pp. 344–345. Duke University, NC (1995)
201. Rough sets: Present state and further prospects. In: T.Y. Lin, A.M. Wildberger (eds.) Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery, pp. 3–5. Simulation Councils, Inc., San Diego, CA (1995)
202. Data versus logic a rough set view. *ICS Research Report*, vol. 28, pp. 1–8. Institute of Computer Science, Warsaw University of Technology (1996)
203. Rough sets: Present state and further prospects. Intelligent Automation and Soft Computing **2**(2), 95–101 (1996)
204. (with J.W. Grzymała-Busse, W. Ziarko and R. Słowiński) Rough Sets. Communication of the ACM **38**(11), 88–95 (1995)
205. Data analysis with rough sets. In: Proceesings of CODATA'96, October, Tsukuba, Japan, (1996)
206. Rough sets, rough relations and rough functions. Fundamenta Informaticae **27**(2-3), 103–108 (1996) (special issue edited by W. Ziarko)
207. Data versus logic a rough set view. In: S. Tsumoto, S. Kobayashi, T. Yokomori, H. Tanaka, A. Nakamura (eds.) Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96), November 6-8, 1996, The University of Tokyo (1996) 1–8
208. Rough sets: Present state and perspectives. In: Proceedings of the Sixth International Conference, Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96), July 1-5, 1996, vol. 2, pp. 1137–1145. Granada, Spain (1996)
209. Some remarks on explanation of data and specification of processes. In: S. Tsumoto (ed.) Bulletin of International Rough Set Society **1**(1), 1–4 (1996)
210. Why rough sets? In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96), September 8-11, 1996, New Orleans, Louisiana, pp. 738–743. IEEE Society Press (1996)
211. Rough sets and data analysis. In: Y.Y. Chen, K. Hirota, J.Y. Yen (eds.) Proceedings of 1996 Asian Fuzzy Systems Symposium Soft Computing in Intelligent Systems and Information Processing, December 11-14, 1996, Kenting, Taiwan, ROC., pp. 1–6. IEEE Society Press (1996)
212. (with T. Munakata) Rough control: Application of rough set theory to control. In: Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing (EUFIT'96), September 2-5, 1996, pp. 209–218. Verlag Mainz, Germany (1996)

213. (with A. Skowron) Helena Rasiowa and Cecylia Rauszer Research on logical foundations of computer science. In: A. Skowron (ed.), Logic, Algebra and Computer Science, Helena Rasiowa and Cecylia Rauszer in Memoriam, Bulletin of the Section of Logic **25**(3-4), 174–184 (1996)
214. Rough set approach to knowledge-based decision support. European Journal of Operational Research **99**(1), (1997) 48–57
215. Rough set and data mining. In: Proceedings of International Conference on Intelligent Processing and Manufacturing Materials, Gold Coast, Australia, pp. 1–5. (1997)
216. Rough sets. In: T.Y. Lin and N. Cercone (eds.) Rough Sets and Data Mining. Analysis of Imprecise Data, pp. 3–7. Kluwer Academic Publishers, Boston, Dordrecht (1997)
217. Rough real functions and rough controllers. In: T.Y. Lin and N. Cercone (eds.), Rough Sets and Data Mining. Analysis of Imprecise Data, Kluwer Academic Publishers, pp. 139–147. Boston, Dordrecht (1997)
218. Conflict analysis. In: Proceedings of the Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), September 9-11, Aachen, pp. 1589–1591. Verlag Mainz, Germany (1997)
219. Vagueness - A rough set view. In: Structures in Logic and Computer Science, Lecture Notes in Computer Science 1261, pp. 106–117. Springer-Verlag, Heidelberg (1997)
220. Rough sets theory and its applications to data analysis. Cybernetics and Systems **29** 661–688 (1998)
221. (with A.G. Jackson and S.R. LeClair) Rough sets applied to discovery of materials knowledge. Journal of Alloys and Compounds **279**, 14–21 (1998)
222. Rough set elements. In: A. Skowron, L. Polkowski (eds.), Rough Sets in Knowledge Discovery, vol. 1. Methodology and Applications, *Studies in Fuzziness and Soft Computing*, vol. 18, pp. 10–30. Springer-Verlag, Hidelberg (1998)
223. An inquiry into anatomy of conflicts. Journal of Information Sciences **109**, 65–78 (1998)
224. Sets, fuzzy sets and rough sets. In: W. Brauer (ed.) Proceedings of Fuzzy-Neuro Systems - Computational Intelligence, March 18-20, 1998, pp. 1–9. München, Germany (1998)
225. Reasoning about data - A rough set perspective. In: A Skowron,L. Polkowski (eds.), Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'98), Warsaw, Poland, June 22-26, 1998, *Lecture Notes in Computer Science*, vol. 1424, pp. 25–34. Springer-Verlag, Heidelberg (1998)
226. Rough modus ponens. In: Proceedings of The Seventh Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems vol. 2, pp. 1162–1166. La Sorbone, Paris (1998)
227. Granularity of knowledge, indiscernibility and rough sets. In: Proceedings of the IEEE Conference on Evolutionary Computation, May 5-9, 1998, Anchorage, Alaska (1998), pp. 106–110. (see also: IEEE Transactions on Automatic Control **XX**, 100–103 (1999))
228. Rough set theory for intelligent industrial applications. In: J. Meech, M. Veiga, M. Smith, S. LeClair (eds.), Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials, pp. 37–44. Honolulu, Hawaii (1999)
229. Data mining - A rough set perspective. In: N. Zhong and L Zhou (eds.), Methodologies for Knowledge Discovery and Data Mining. Proceedings of the Third Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99), April 26-28, 1999, Beijing, China, *Lecture Notes in Artificial Intelligence*, vol. 1574, pp. 3–12. Springer-Verlag, Heidelberg (1999)
230. Rough sets, rough functions and rough calculus. In: S.K. Pal, A. Skowron (eds.) Rough Fuzzy Hybridization, A New Trend in Decision Making, pp. 99–109. Springer-Verlag, Singapore (1999)

231. (with J. Komorowski, L. Polkowski and A. Skowron) Rough sets: A tutorial. In: S.K. Pal, and A. Skowron (eds.) Rough Fuzzy Hybridization, A New Trend in Decision Making, pp. 3–98. Springer-Verlag, Singapore (1999)
232. Logic, probability and rough sets. In: J. Karhumaki, H. Maurer, G. Paun and G. Rozenberg (eds.), Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa, pp. 364–373. Springer-Verlag, Heidelberg (1999)
233. Decision rules, Bayes' rule and rough sets. In: N. Zhong. A, Skowron, S. Ohsuga (eds.) Proceedings of the 7th International Workshop on Rough Sets, Fuzzy Sets, Data Mining and Granular-Soft Computing (RSFSGrC 99) Ube-Yamaguchi, Japan, November 9-11, 1999, *Lecture Notes in Artificial Intelligence*, vol. 1711, pp. 1–9. Springer- Verlag, Heidelberg (1999)

2000–2007

234. AI and intelligent industrial applications: The rough set perspective. Cybernetics and Systems **31**(3), 227-252 (2000)
235. Rough sets and decision algorithms. In: W. Ziarko, Y.Y. Yao, (eds.), Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTS'2000), October 16-19, 2000, pp. 1–16. Banff Canada (2000) (see also: *Lecture Notes in Artificial Intelligence*, vol. 2005, pp. 30–45. Springer- Verlag, Heidelberg (2000))
236. Rough sets and decision analysis. INFOR: Information Systems and Operational Research **38**(2), 132–144 (2000)
237. (with: L. Polkowski and A. Skowron). Rough sets and rough logic: A KDD perspective. In: L. Polkowski, S. Tsumoto,T.Y.Lin (eds.) Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems, *Studies in Fuzziness and Soft Computing*, vol. 56, pp. 583–648. Physica-Verlag, Springer, Heidelberg (2000)
238. Drawing conclusions from data - The rough set way. International Journal of Intelligent Systems **16**(1), 3–11 (2001) (special issue: Rough Set Approach to Reasoning About Data A. Skowron, J.F. Peters (eds.))
239. Rough sets and their applications. In: B. Reusch and K-H. Theme (eds.) Computational Intelligence in Theory and Practice, Advances in Soft Computing, pp. 75–93. Physica-Verlag, Springer-Verlag, Heidelberg (2001)
240. New look on Bayes' theorem the rough set outlook. In: S. Hirano, M. Inuigucgi, S. Tsumoto (eds.) Proceedings of International Workshop on Rough Set Theory and Granular Computing (RSTGC 2001) Bulletin of International Rough Set Society **5**(1-2), 1–8 (2001) (see also: Bayes' theorem revisited -The rough set view. In: T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa and T. Washio (eds.), New Frontiers in Artificial Intelligence, *Lecture Notes in Artificial Intelligence*, vol. 2253, pp. 240–250. Springer-Verlag, Berlin (2001))
241. Data analysis: The rough set perspective. In: Zbiory rozmyte i ich zastosowania (Fuzy Sets and their Applications)). pp. 173–181. Gliwice (2001) (in Polish)
242. Combining rough sets and Bayes' rule. Computational Intelligence **17**(3), 401–409 (2001)
243. Rough sets, decision algorithms and Bayes' theorem. European Journal of Operational Research **136**, 181–189 (2002)
244. Decision tables and decision spaces. In: Proceedings of the Sixth International Conference on Soft Computing and Distributed Systems (SCDO'02), pp. 7–11. Rzeszów, Poland

245. In pursuit of patterns in data. Reasoning from data the rough set way. In: J. J. Alpigini, J. F. Peters, A. Skowron, N. Zhong, J. Liu (eds.) Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'02), October 14-16, 2002, Malvern, PA, USA, *Lecture Notes in Artificial Intelligence*, vol. 2475, pp. 1–9. Springer-Verlag, Heidelberg (2002)
246. (with S. Grecco, R. Słowiński) Generalized decision algorithms, rough inference rules and flow graphs. In: J. J. Alpigini, J. F. Peters, A. Skowron, N. Zhong, J. Liu (eds.) Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'02), October 14-16, 2002, Malvern, PA, USA, *Lecture Notes in Artificial Intelligence*, vol. 2475, pp. 95–104. Springer-Verlag, Heidelberg (2002)
247. A primer on rough sets: A new approach to drawing conclusions from data. Cardozo Law Review **22**(5-6), 1407–1415 (2001) (see also: A primer on rough sets: A new approach to drawing conclusions from data. In: M. MacCrimon, P.Tillers (eds.), The Dynamics of Judicial Proof, pp. 135–144. Springer-Verlag, Heidelberg (2002))
248. Granularity, multivalued logic, Bayes' theorem and rough sets. In: T.Y. Lin, Y.Y. Yao, L. Zadeh (eds.) Data mining, rough sets and granular computing, pp. 487–499. Elsevier, Amsterdam (2002)
249. (with A. Skowron) Rough set rudiments. Bulletin of the International Rough Set Society **6**(2), 27–31 (2002)
250. (with A. Skowron, J. Komorowski, L. Polkowski) Rough sets perspective on data and knowledge. In: W. Klösgen and J. Żytkow (eds.), Handbook of Data Mining and Knowledge Discovery, pp. 134–149. Oxford University Press (2002)
251. Theorize with data using rough sets. In: Proceedings of 26th Annual International Computer Software and Applications Conference (IEEE COMPSAC'02), August 2002, Oxford, England, pp. 1125–1129. IEEE Society Press (2002)
252. Rough sets and intelligent data analysis. Information Sciences **147**, 1–12 (2002)
253. The rough set view on Bayes' theorem. In: N. R. Pal and M. Sugeno (eds.) Proceedings of the International Conference on Fuzzy Systems (AFSS'02), February 3-6, 2002, Calcutta, India, *Lecture Notes in Computer Science*, vol. 2275, pp. 106–116. Springer-Verlag, Heidelberg (2002)
254. Podziękowanie (Thanks). In: K. Długosz (red.). Zdzisław Pawlak. Doctor Honoris Causa Politechniki Poznańskiej, pp. 15–18. Politechnika Poznańska, Poznań (2002) (in Polish)
255. I co dalej? (And what is the next?). In: K. Długosz (red.). Zdzisław Pawlak. Doctor Honoris Causa Politechniki Poznańskiej, pp. 21–29. Politechnika Poznańska, Poznań (2002) (in Polish)
256. A rough set view on Bayes' theorem. International Journal of Intelligent Systems **18**(5), 487–498 (2003)
257. Elementary rough set granules: Toward a rough set processor. In: S. K. Pal, L. Polkowski, A. Skowron, (eds.), Rough-Neurocomputing: Techniques for Computing with Words, *Cognitive Technologies*, pp. 5–14. Springer-Verlag, Berlin (2004)
258. Bayes' theorem the rough set perspective. In: M. Inuiguchi, S. Tsumoto and S. Hirano (eds.), Rough Set Theory and Granular Computing, *Series in Fuzziness and Soft Computing*, vol.125, pp. 1–12. Springer-Verlag, Heidelberg (2003)
259. (with J. Peters, A. Skowron, Z. Suraj, S., Ramanna) Rough measures, rough integrals and sensor fusion. In: M. Inuiguchi, S. Hirano, S. Tsumoto (eds.), Rough Set Theory and Granular Computing, *Series in Fuzziness and Soft Computing*, vol. 125, pp. 263–272. Springer-Verlag, Heidelberg (2003)

260. Foreword. In: M. Inuiguchi, S. Hirano, S. Tsumoto (eds.), Rough Set Theory and Granular Computing, *Series in Fuzziness and Soft Computing*, vol. 125, pp. 1. Springer-Verlag, Heidelberg (2003)
261. Flow graphs and decision algorithms. In: G. Wang, Q. Liu, Y. Yao and A. Skowron, (eds.), Proceedings of the International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'03), Chongqing, China, May 26-29, 2003, *Lecture Notes in Artificial Intelligence*, vol. 2639, pp. 1–15. Springer-Verlag, Heidelberg (2003)
262. Probability truth and flow graphs. Electronic Notes in Theoretical Computer Science **82**(4), 1–9 (2003). (A. Skowron, M. Szczuka (eds.) Proceedings of the International Workshop on Rough Sets in Knowledge Discovery and Soft Computing at ETAPS 2003, April 12-13, 2003,Warsaw, Poland)
263. Decision rules and dependencies. In: L. Czaja (ed.) Proceedings of the Workshop on Concurrency, Specification & Programming (CS&P'03) vol. 1, September 2003, Czarna, Poland, pp. 35–40. Warsaw University (2003)
264. Decision algorithms and flow graphs: A rough set approach. Journal of Telecommunications and Information Technology **3**, 98–101 (2003)
265. Rough sets, Bayes' theorem and flow graphs. In: B. Bouchon-Meunier, L. Foullo and R. Yager, (eds.), Proceedings of the Ninth International Conference Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'02), July 1-5, 2002, pp. 1619–1624. Annecy, France, (see also: Rough sets, Bayes' theorem and flow graphs. In: B. Bouchon-Meunier, L. Foulloy and R.R. Yager (eds.) Intelligent Systems for Information Processing: From Representation to Applications, pp. 243–252. Elsevier (2003))
266. Data analysis and flow graphs. In: Proceedings of the Third Internal Conference on Decision Support for Telecommunications and Information Society, Warsaw, Poland September 2003, pp. 137–144. (see also: Journal of Telecommunications and Information Technology **3**, 1–5 (2004))
267. Data mining and flow graphs. In: 17th JISR-IIASAWorkshop on Methodologies and Tools for Complex System Modeling and Integrated Policy Assessment (CSM 2003), September, 8 - 10, International Institute for Applied Systems Analysis, pp. 49–50. Laxenburg, Austria (2003)
268. Decision rules and flow networks. European Journal of Operational Research **154**, 184–190 (2004)
269. Decision networks. In: S. Tsumoto, R. Słowiński, J. Komorowski, J. W. Grzmała-Busse (eds.), Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'04), June 1-5, 2004, Uppsala, Sweden, *Lecture Notes in Artificial Intelligence*, vol. 3066, pp. 1–7. Springer, Heidelberg (2004)
270. (with S. Greco, R. Słowiński) Bayesian confirmation measures within rough set approach. In: S. Tsumoto, R. Słowiński, J. Komorowski, J. W. Grzmała-Busse (eds.), Proceedings of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'04), June 1-5, 2004, Uppsala, Sweden, *Lecture Notes in Artificial Intelligence*, vol. 3066, pp. 264–273. Springer, Heidelberg (2004)
271. Zbiory przybliżone, nowa matematyczna metoda analizy danych – Na podstawie odczytu wygłoszonego w dniu 30 października 2003 roku. (Rough sets, a new mathematical method of data analysis – On the basis of lecture delivered on October 30, 2003) In: S. Janeczko (ed.), Konserwatorium Politechniki Warszawskiej, Wkładka do Miesięcznika Politechniki Warszawskiej **5**, 1–13 (2004) (in Polish)

272. Inference rules and decision rules. In: L. Rutkowski, J. Siekman, R. Tadeusiewicz, L. Zadeh (eds.), Proceedings of the 7th International Conference on Artificial Intelligence and Soft Computing (ICAISC'04), June 7-11, 2004, Zakopane, Poland, *Lecture Notes in Artificial Intelligence*, vol. 3070, pp. 102–108. Springer-Verlag, Heidelberg (2004)
273. (with S. Greco, R. Słowiński) Can bayesian confirmation measures be useful for rough set decision rules? Artificial Intelligence, Engineering Applications of Artificial Intelligence **17**, 345–361 (2004)
274. (with J. Peters and A. Skowron) Approximating functions with rough sets. In: S. Dick, L. Kurgan, P. Misilek, W. Pedrycz, M. Reformat (eds.), Proceedings 2004 of the IEEE North American Fuzzy Information Society (NAFIPS'04) June 27-30, 2004, Banff, Alberta, Canada, vol. No. 04TH8736C, pp. 785–790. IEEE Computer Press, Los Alamitos, CA (2004)
275. Decision rules and dependencies. Fundamenta Informaticae **60**(1-4), 33–39 (2004)
276. Some issues on rough sets. Transactions on Rough Sets **I**, *LNCS*, vol. 3100, pp. 1–58. Springer, Heidelberg (2004)
277. Flow graphs - A new paradigm for data mining and knowledge discovery. Proceedings of the JAIST Forum 2004 - Technology Creation Based on Knowledge Science: Theory and Practice, jointly with 5th International Symposium on Knowledge and Systems Science (KSS'04), pp. 147–153. Japan Advanced Institute of Science and Technology (JAIST) (2004)
278. Flow graphs and data mining. Transactions on Rough Sets **III**, *LNCS*, vol. 3400, pp. 1–36. Springer, Heidelberg (2005)
279. Flow graphs, their fusion and data analysis. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, M. Szczuka (eds.), Monitoring, Security, and Rescue Techniques in Multiagent Systems, *Advances in Soft Computing*, pp. 1–10. Springer-Verlag, Heidelberg (2005)
280. Orthodox and non-orthodox sets - Some philosophical remarks. Foundations of Computing and Decision Sciences **30**(2), 133–140 (2005)
281. A treatise on rough sets. Transactions on Rough Sets **IV**, *LNCS*, vol. 3700, pp. 1–17. Springer, Heidelberg (2005)
282. Rough sets and flow graphs. In: D. Ślęzak, G. Wang, M.,Szczuka, I. Duentsch, Y.Y. Yao (eds.), Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'05), September 1-3, 2005, Regina, Canada, *Lecture Notes in Artificial Intelligence*, vol. 3641, pp. 1–11. Springer, Heidelberg (2005)
283. Some remarks on conflict analysis. European Journal of Operational Research **166**(3), 649–654 (2005)
284. (with L. Polkowski, A. Skowron) Rough sets: An approach to vagueness. In: L.C. Rivero, J.H. Doorn, V.E. Ferraggine (eds.), Encyclopedia of Database Technologies and Applications, pp. 575–580. Idea Group Inc., Hershey, PA (2005)
285. Conflicts and negotiations. In: G. Wang, J.F. Peters, A. Skowron, Y.Y. Yao (eds.), Proceedings of the International Conference on Rough Sets and Knowledge Technology (RSKT'06), Chongqing, China, July 24-26, 2006, *Lecture Notes in Artificial Intelligence*, vol. 4062, pp. 12–27. Springer, Heidelberg (2006)
286. Decision trees and flow graphs. In: S. Greco, Y. Hata, S. Hirano, M. Inuiguchi, S. Miyamoto, H.S. Nguyen. R. Słowiński (eds.) Proceedings of the Fifth International Conference on Rough Sets and Current Trends in Computing (RSCTC'06), November 6-8, 2006, Kobe, Japan, *Lecture Notes in Artificial Intelligence*, vol. 4259, pp. 1–11. Springer, Heidelberg (2006)

287. (with A. Skowron) Rough sets and conflict analysis. In: Jie Lu, Da Ruan, Guangquan Zhang (eds.), E-Service Intelligence Methodologies, Technologies and Applications, *Studies in Computational Intelligence*, vol. 37, pp. 35–74. Springer, Heidelberg (2006)
288. (with A. Skowron) Rudiments of rough sets. Information Sciences **177**(1), 3–27 (2007)
289. (with A. Skowron) Rough sets: Some extensions. Information Sciences **177**(1), 28–40 (2007)
290. (with A. Skowron). Rough sets and Boolean reasoning. Information Sciences **177**(1), 41–73 (2007)

Manuscripts and Articles in Newspapers

1. Organizacja maszyn matematycznych. Cześć I: Pojęcie maszyny matematycznej. Komentarz do wykładu dla studentów matematyki Wydziału Matematyki i Mechaniki Uniwersytetu Warszawskiego w latach 1963-1967. (Computer organization. Part I: Concept of computer. Lecture notes for students of mathematics in Faculty of Mathematics and Mechanics at Warsaw University, 1963-1967), pp. 1–52 (in Polish)
2. Organizacja maszyn matematycznych. Cześć II Maszyny uniwersalne. Komentarz do wykładu dla studentów matematyki na Wydziale Matematyki i Mechaniki Uniwersytetu Warszawskiego w latach 1963-1967. (Computer organization. Part II Universal computers. Lecture notes for students of mathematics in Faculty of Mathematics and Mechanics at Warsaw University, 1963-1967), pp. 1–52 (in Polish)
3. Teoria maszyn matematycznych, Zeszyt 1. Komentarz do wykładu doc. dr Zdzisława Pawlaka dla III-go roku Studium Zaocznego Matematyki UW, Sekcja Metod Numerycznych, Wydział Matematyki i Mechaniki Uniwersytetu Warszawskiego w latach 1963-1967. (Theory of computers. Lecture notes of doc. dr Zdzisław Pawlak for students of the 3rd year in Faculty of Mathematics and Mechanics at Warsaw University, 1963-1967), pp. 1–145 (in Polish)
4. Rough set rudiments. pp. 1–47 (manuscript)
5. Zbiory przybliżone. Nowa matematyczna metoda analizy danych. (Rough sets. New mathematical method of data analysis) pp. 1–13 (in Polish)
6. Rough control. (1996) pp. 1–4 (manuscript)
7. Komputery i nauka (Computers and science). pp. 1–7 (manuscript) (in Polish)
8. Rough sets - A new paradigm of imprecise knowledge. pp. 1–7 (manuscript)
9. Conflicts and decisions II. pp. 1–7 (manuscript)
10. Data analysis: A rough set view. pp. 1–9 (manuscript)
11. Data, decision rules and rough sets. pp. 1–10 (manuscript)
12. Decision and flow networks. pp. 1–8 (manuscript)
13. Decision tables and case based reasoning. pp. 1–5 (manuscript)
14. Flow graphs: A new paradigm for intelligent data analysis. pp. 1–28 (manuscript)
15. Inference rules, decision rules and rough sets. (1999) pp. 1–7 (manuscript, preliminary version of the paper for the RSFDGrC 1999 lecture in Yamaguchi).
16. Multi-valued logic, Bayes' rule and rough sets. pp. 1–10 (manuscript)
17. A new approach to drawing conclusions from data. A rough set perspective. pp. 1-14 (manuscript)
18. New look on Bayes' theorem the rough set outlook II. pp. 1-17 (manuscript)
19. Rough sets - A new way of data analysis. pp. 1-9 (manuscript)
20. Rough set theory: A new mathematical approach to data analysis. pp. 1–14 (manuscript)

21. Using rough sets for drawing conclusions from data. pp. 1–8 (manuscript)
22. On some issues on rough sets. (2004) pp. 1–60 (manuscript)
23. Rough controllers. pp. 1–6 (manuscript)
24. Conflicts and decisions. pp. 1–6 (manuscript)
25. Rough sets. pp. 1–40 (slides)
26. In pursuit of patterns in data. Reasoning from data. The rough set way. pp. 1–36 (slides).
27. Zbiory przybliżone i ich zastosowania (Rough sets and their applications). pp. 1–4 (manuscript, in Polish)
28. Rough set theory. pp. 1–5 (manuscript)
29. (with L. Łukaszewicz, J. Fiett, W. Jaworski, Z. Sawicki) W sprawie polskich maszyn matematycznych (On Polish computers). Trybuna Ludu 131, p. 4 (1956) (article in Polish)
30. Maszyny bezadresowe i 'symbolika polska' Łukasiewicza (Address-less machines and 'Polish notation' of Łukasiewicz). Życie Warszawy 113, p. 5 (1961) (article in Polish)
31. Matematyczne podstawy informatyki (Mathematical foundations of informatics). Trybuna Ludu 105, p. 6 (1973) (article in Polish)
32. Badania dla informatyki (Research in informatics). Trybuna Ludu 36, p. 8 (1974) (article in Polish)
33. Ludzie i komputery. Kształcenie informatyków (Humans and computers. Education in informatics). Trybuna Ludu 345. p. 6 (1974) (article in Polish)
34. Wracanie do źródeł (Returning to the roots). Głos Pracy 17, p. 1 (wtorek, 25 lipca 1978) (article in Polish)

Chapter 3
Rough Sets: From Rudiments to Challenges

Hung Son Nguyen and Andrzej Skowron

Abstract. In the development of rough set theory and applications, one can distinguish three main stages. At the beginning, the researchers concentrated on descriptive properties such as reducts of information systems preserving indiscernibility relations or description of concepts or classifications. Next, they moved to applications of rough sets in machine learning, pattern recognition and data mining. After gaining some experiences, they developed foundations for inductive reasoning leading to, *for example*, inducing classifiers. While the first period was based on the assumption that objects are perceived by means of partial information represented by attributes, the second period was based on the assumption that information about the approximated concepts is partial too. Approximation spaces and searching strategies for relevant approximation spaces were recognized as the basic tools for rough sets. Important achievements both in theory and applications were obtained using Boolean reasoning and approximate Boolean reasoning applied, *for example*, in searching for relevant features, discretization, symbolic value grouping, or, in more general sense, in searching for relevant approximation spaces. Nowadays, we observe that a new period is emerging in which two new important topics are investigated: (i) strategies for discovering relevant (complex) contexts of analysed objects or granules, what is strongly related to information granulation process and granular computing, and (ii) interactive computations on granules. Both directions are aiming at developing tools for approximation of complex vague concepts, such as behavioural patterns or adaptive strategies, making it possible to achieve the satisfactory qualities of realized interactive computations. This chapter presents this development from rudiments of rough sets to challenges, *for example*, related to ontology approximation, process mining, context inducing or Perception-Based Computing (PBC). The approach is based on Interactive Rough-Granular Computing (IRGC).

Keywords: vague concept, indiscernibility, reduct, approximation space, rough sets, decision rule, dependency, (approximate) Boolean reasoning and rough sets, concept approximation, ontology approximation, scalability in data mining, (interactive rough) granular computing, context inducing, process mining, perception-based computing.

Hung Son Nguyen · Andrzej Skowron
Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
e-mail: skowron@mimuw.edu.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 75–173.
springerlink.com

3.1 Introduction

Rough set theory, proposed by Professor Zdzisław Pawlak in 1982 [209, 211, 212, 215], can be seen as a new mathematical approach for solving problems dealing with imperfect data and knowledge or, in particular, with vague concepts.

The rough set philosophy is founded on the assumption that with every object of the universe of discourse, we associate some information (data, knowledge). For example, if objects are patients suffering from a certain disease, symptoms of the disease form information about patients. Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory. This understanding of indiscernibility is related to the idea of Gottfried Wilhelm Leibniz that objects are indiscernible if and only if all available functionals take on identical values (Leibniz's Law of Indiscernibility: The Identity of Indiscernibles) [5, 127]. However, in the rough set approach, indiscernibility is defined relative to a given set of functionals (attributes).

Any set of all indiscernible (similar) objects is called an elementary set, and forms a basic granule (atom) of knowledge about the universe. Any union of elementary sets is referred to as a crisp (precise) set[1]. A set that is not crisp is called rough (imprecise or vague).

Consequently, each rough set has boundary region cases, that is, objects that cannot with certainty be classified either as members of the set or of its complement. Obviously, crisp sets have no boundary region elements at all. This means that boundary region cases cannot be properly classified by employing available knowledge.

Thus, the assumption that objects can be "seen" only through the information available about them leads to the view that knowledge has a granular structure. Due to the granularity of knowledge, some objects of interest cannot be discerned and appear as the same (or similar). As a consequence, vague concepts, in contrast to precise concepts, cannot be characterized in terms of information about their elements. Therefore, in the proposed approach, we assume that any vague concept is replaced by a pair of precise concepts – called the lower and the upper approximation of the vague concept. The lower approximation consists of all objects that surely belong to the concept and the upper approximation contains all objects that possibly belong to the concept. The difference between the upper and the lower approximation constitutes the boundary region of the vague concept. These approximations are two basic operations in rough set theory. Note, that the boundary region is defined relative to a subjective knowledge given by a set of attributes or/and sample of objects. Such a boundary region is crisp. However, when some attributes are deleted, new attributes are added or a given sample is updated and the boundary region is changing. One could ask about a boundary region independent of such subjective knowledge but then, in the discussed framework, we do not have a possibility to define such

[1] This approach is generalized when one considers inductive extensions of approximations from samples of objects (see, *e.g.* [303]).

region as a crisp set. This property is related to the higher-order vagueness discussed in philosophy.

Hence, rough set theory expresses vagueness not by means of membership, but by employing a boundary region of a set. If the boundary region of a set is empty, it means that the set is crisp, otherwise the set is rough (inexact). A nonempty boundary region of a set means that our knowledge about the set is not sufficient to define the set precisely.

Rough set theory is not an alternative to but rather is embedded in classical set theory. Rough set theory can be viewed as a specific implementation of Frege's idea of vagueness [59], that is, imprecision in this approach is expressed by a boundary region of a set.

Rough set theory has attracted worldwide attention of many researchers and practitioners, who have contributed essentially to its development and applications. Rough set theory overlaps with many other theories. Despite this, rough set theory may be considered as an independent discipline in its own right. The rough set approach seems to be of fundamental importance in artificial intelligence and cognitive sciences, especially in research areas such as machine learning, intelligent systems, inductive reasoning, pattern recognition, mereology, image processing, signal analysis, knowledge discovery, decision analysis and expert systems. The main advantage of rough set theory in data analysis is that it does not need any preliminary or additional information about data like probability distributions in statistics, basic probability assignments in Dempster–Shafer theory, a grade of membership or the value of possibility in fuzzy set theory (see, *e.g.* [55] where some combinations of rough sets with non-parametric statistics are studied). One can observe the following about the rough set approach:

- introduction of efficient algorithms for finding hidden patterns in data,
- determination of optimal sets of data (data reduction),
- evaluation of the significance of data,
- generation of sets of decision rules from data,
- easy-to-understand formulation,
- straightforward interpretation of obtained results,
- suitability of many of its algorithms for parallel processing.

The basic ideas of rough set theory and its extensions as well as many interesting applications can be found in a number of books (see, *e.g.* [43, 46, 52, 55, 83, 104, 118, 119, 137, 189, 199, 200, 215, 245, 249, 252, 253, 277, 324, 387, 109, 156, 38, 96, 223, 45, 29, 110, 198, 159, 248, 144, 224]), issues of the Transactions on Rough Sets [233, 231, 226, 227, 228, 232, 235, 229, 236, 241, 230, 238, 11, 237], special issues of other journals (see, *e.g.* [37, 133, 225, 197, 291, 326, 391, 392, 105, 191, 44, 292]), proceedings of international conferences (see, *e.g.* [2, 99, 136, 251, 290, 310, 320, 321, 353, 355, 356, 367, 390, 394, 72, 368, 4, 377, 123, 39, 366, 370, 272, 350, 384, 124, 378]) and tutorials (see, *e.g.* [117, 222, 221, 220]). For more information on the bibliography on rough sets, one can also visit web pages[2].

[2] www.roughsets.org

In this chapter, we begin with a short discussion on vague concepts (see Section 3.2). Next, we recall the basic concepts of rough set theory (see Section 3.3). Some extensions of the rough set approach are outlined in Section 3.4. In Section 3.5, we discuss the relationship of the rough set approach with inductive reasoning. In particular, we present the rough set approach to inducing rough set-based classifiers and inducing relevant approximation spaces. We also discuss shortly the relationship of the rough set approach and the higher-order vagueness. Section 3.6 includes some remarks on relationships of information granulation and rough sets. Section 3.7, we outline the rough set approach to ontology approximation. The rough set approach based on combination of rough sets and Boolean reasoning with applications in pattern recognition, machine learning and data mining in presented in Section 3.8. In Section 3.9, we discuss some scalability issues using the rough set approach. Some comments on relationships of rough sets and logic are included in Section 3.10. Finally, we discuss some challenging issues for rough sets (see Section 3.11). We propose Interactive (Rough) Granular Computing (IRGC) as a framework making it possible to search for solutions of problems related to inducing of relevant contexts, process mining and Perception-Based Computing (PBC).

This chapter is an extended version of our paper presented in the book "Three Approaches to Data Analysis. Test Theory, Rough Sets and Logical Analysis of Data" [40].

3.2 Vague Concepts

Mathematics requires that all mathematical notions (including set) must be exact, otherwise precise reasoning would be impossible. However, philosophers [111, 112, 266, 271] and recently computer scientists [145, 184, 186, 287] as well as other researchers have become interested in *vague* (imprecise) concepts.

In classical set theory, a set is uniquely determined by its elements. In other words, this means that every element must be uniquely classified as belonging to the set or not. That is to say the notion of a set is a *crisp* (precise) one. For example, the set of odd numbers is crisp because every number is either odd or even.

In contrast to odd numbers, the notion of a beautiful painting is vague, because we are unable to classify uniquely all paintings into two classes: *beautiful* and not *beautiful*. Some paintings cannot be decided whether they are beautiful or not, and thus, they remain in the doubtful area. Thus, *beauty* is not a precise but a vague concept.

Almost all concepts we are using in natural language are vague. Therefore, common-sense reasoning based on natural language must be based on vague concepts and not on classical logic. Interesting discussion of this issue can be found in [266].

The idea of vagueness can be traced back to the ancient Greek philosopher Eubulides of Megara (ca. 400BC) who first formulated so-called sorites (heap) and falakros (bald man) paradoxes (see, *e.g.* [111, 112]). The bald man paradox goes as

follows: suppose a man has 100,000 hairs on his head. Removing one hair from his head surely cannot make him bald. Repeating this step we arrive at the conclusion that a man without any hair is not bald. Similar reasoning can be applied to a heap of stones.

Vagueness is usually associated with the boundary region approach (i.e. existence of objects that cannot be uniquely classified relative to a set or its complement) which was first formulated in 1893 by the father of modern logic, German logician, Gottlob Frege (1848-1925) (see [59]).

According to Frege, the concept must have a sharp boundary. To the concept without a sharp boundary, there would correspond an area that would not have any sharp boundary–line all around. It means that mathematics must use crisp, not vague concepts, otherwise it would be impossible to reason precisely.

Summing up, vagueness is

- not allowed in mathematics;
- interesting for philosophy;
- a nettlesome problem for natural language, cognitive science, artificial intelligence, machine learning, philosophy, and computer science.

3.3 Rudiments of Rough Sets

This section briefly delineates basic concepts in rough set theory.

3.3.1 Indiscernibility and Approximation

The starting point of rough set theory is the indiscernibility relation, which is generated by information about objects of interest. The indiscernibility relation expresses the fact that due to a lack of information (or knowledge) we are unable to discern some objects employing available information (or knowledge).

This means that, in general, we are unable to deal with each particular object but we have to consider granules (clusters) of indiscernible objects as a fundamental basis for our theory.

From a practical point of view, it is better to define basic concepts of this theory in terms of data. Therefore, we will start our considerations from a data set called an *information system*. An information system is a data table containing rows labeled by objects of interest, columns labelled by attributes and entries of the table are attribute values. For example, a data table can describe a set of patients in a hospital. The patients can be characterized by some attributes, like *age, sex, blood pressure and body temperature*. With every attribute, a set of its values is associated, *for example*, values of the attribute *age* can be *young, middle*, and *old*. Attribute values can be also numerical. In data analysis, the basic problem we are interested in is to find patterns in data, that is, to find a relationship between some sets of attributes, *for example*, we might be interested whether *blood pressure* depends on *age and sex*.

Suppose, we are given a pair $\mathbb{A} = (U, A)$ of non-empty, finite sets U and A, where U is the *universe* of *objects* and A is a set consisting of *attributes*, that is, functions $a : U \longrightarrow V_a$ where V_a is the set of values of attribute a, called the *domain* of a. The pair $\mathbb{A} = (U, A)$ is called an *information system* (see, *e.g.* [210]). Any information system can be represented by a data table with rows labelled by objects and columns labelled by attributes[3]. Any pair (x, a), where $x \in U$ and $a \in A$ defines the table entry consisting of the value $a(x)$.

Any subset B of A determines a binary relation IND_B on U called an *indiscernibility relation*, defined by

$$x \; IND_B \; y \text{ if and only if } a(x) = a(y) \text{ for every } a \in B, \tag{3.1}$$

where $a(x)$ denotes the value of attribute a for object x.

Obviously, IND_B is an equivalence relation. The family of all equivalence classes of IND_B, that is, the partition determined by B, will be denoted by U/IND_B, or simply U/B; an equivalence class of IND_B, that is, the block of the partition U/B, containing x will be denoted by $B(x)$ (other notation used: $[x]_B$ or more precisely $[x]_{IND_B}$). Thus, in view of the data we are unable, in general, to observe individual objects but we are forced to reason only about the accessible granules of knowledge (see, *e.g.* [199, 215, 255]).

If $(x, y) \in IND_B$, we will say that x and y are *B-indiscernible*. Equivalence classes of the relation IND_B (or blocks of the partition U/B) are referred to as *B-elementary sets* or *B-elementary granules*. In the rough set approach, the elementary sets are the basic building blocks (concepts) of our knowledge about reality. The unions of *B-elementary sets* are called *B-definable sets*.[4]

For $B \subseteq A$ we denote by $Inf_B(x)$ the *B-signature* of $x \in U$, that is, the set $\{(a, a(s)) : a \in B\}$. Let $INF(B) = \{Inf_B(s) : s \in U\}$. Then, for any objects $x, y \in U$, the following equivalence holds: $x IND_B y$ if and only if $Inf_B(x) = Inf_B(y)$.

The indiscernibility relation will be further used to define basic concepts of rough set theory. Let us define now the following two operations on sets $X \subseteq U$

$$\mathsf{LOW}_B(X) = \{x \in U : B(x) \subseteq X\}, \tag{3.2}$$

$$\mathsf{UPP}_B(X) = \{x \in U : B(x) \cap X \neq \varnothing\}, \tag{3.3}$$

assigning to every subset X of the universe U two sets $\mathsf{LOW}_B(X)$ and $\mathsf{UPP}_B(X)$ called the *B-lower* and the *B-upper approximation* of X respectively. The set

$$\mathsf{BN}_B(X) = \mathsf{UPP}_B(X) - \mathsf{LOW}_B(X), \tag{3.4}$$

will be referred to as the *B-boundary region* of X.

[3] Note that in statistics or machine learning such a data table is called a sample [97].

[4] One can compare data tables corresponding to information systems with relations in relational databases [63].

From the definition, we obtain the following interpretation:

- The *lower approximation* of a set X with respect to B is the set of all objects that can be for *certain* classified as objects in X using B (are *certainly* in X in view of B).
- The *upper approximation* of a set X with respect to B is the set of all objects that can be *possibly* classified as objects in X using B (are *possibly* in X in view of B).
- The *boundary region* of a set X with respect to B is the set of all objects, that can be classified neither as in X nor as in $U - X$ using B.

In other words, due to the granularity of knowledge, rough sets cannot be characterized by using available knowledge. Therefore with every rough set we associate two *crisp* sets, called *lower* and *upper approximation*. Intuitively, the lower approximation of a set consists of all elements that *surely* belong to the set, whereas the upper approximation of the set constitutes of all elements that *possibly* belong to the set, and the *boundary region* of the set consists of all elements that cannot be classified uniquely to the set or its complement, by employing available knowledge. The approximation definition is clearly depicted in Figure 3.1.

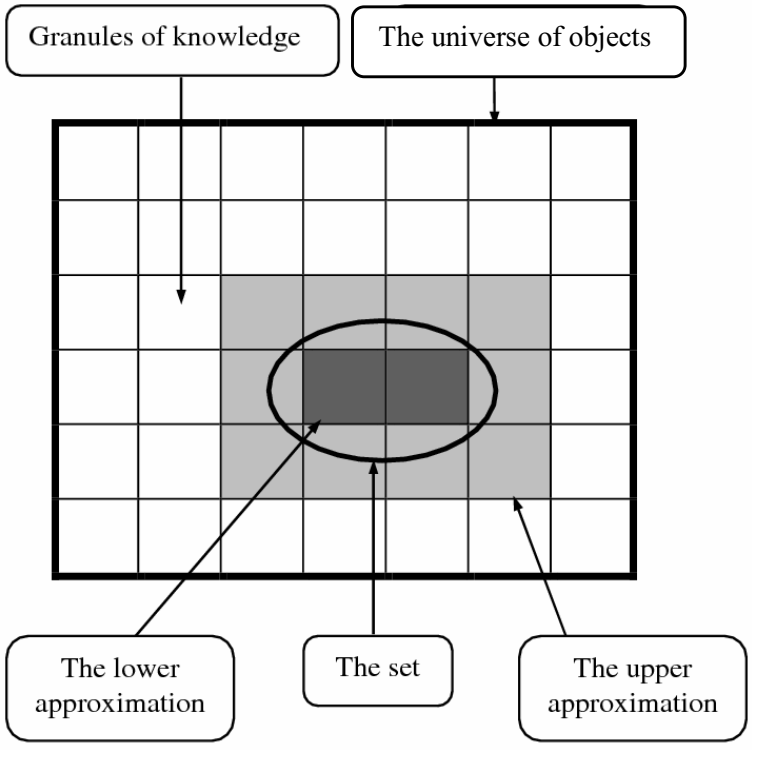


Fig. 3.1. A rough set

The approximations have the following properties:

$$\begin{aligned}
&\mathsf{LOW}_B(X) \subseteq X \subseteq \mathsf{UPP}_B(X), \qquad (3.5)\\
&\mathsf{LOW}_B(\varnothing) = \mathsf{UPP}_B(\varnothing) = \varnothing, \mathsf{LOW}_B(U) = \mathsf{UPP}_B(U) = U,\\
&\mathsf{UPP}_B(X \cup Y) = \mathsf{UPP}_B(X) \cup \mathsf{UPP}_B(Y),\\
&\mathsf{LOW}_B(X \cap Y) = \mathsf{LOW}_B(X) \cap \mathsf{LOW}_B(Y),\\
&X \subseteq Y \text{ implies } \mathsf{LOW}_B(X) \subseteq \mathsf{LOW}_B(Y) \text{ and } \mathsf{UPP}_B(X) \subseteq \mathsf{UPP}_B(Y),\\
&\mathsf{LOW}_B(X \cup Y) \supseteq \mathsf{LOW}_B(X) \cup \mathsf{LOW}_B(Y),\\
&\mathsf{UPP}_B(X \cap Y) \subseteq \mathsf{UPP}_B(X) \cap \mathsf{UPP}_B(Y),\\
&\mathsf{LOW}_B(-X) = -\mathsf{UPP}_B(X),\\
&\mathsf{UPP}_B(-X) = -\mathsf{LOW}_B(X),\\
&\mathsf{LOW}_B(\mathsf{LOW}_B(X)) = \mathsf{UPP}_B(\mathsf{LOW}_B(X)) = \mathsf{LOW}_B(X),\\
&\mathsf{UPP}_B(\mathsf{UPP}_B(X)) = \mathsf{LOW}_B(\mathsf{UPP}_B(X)) = \mathsf{UPP}_B(X).
\end{aligned}$$

Let us note that the inclusions in (3.5) cannot be in general substituted by the equalities. This has some important algorithmic and logical consequences.

Now we are ready to give the definition of rough sets.

If the boundary region of X is the empty set, that is, $\mathsf{BN}_B(X) = \varnothing$, then the set X is *crisp* (*exact*) with respect to B; in the opposite case, that is, if $\mathsf{BN}_B(X) \neq \varnothing$, the set X is referred to as *rough* (*inexact*) with respect to B. Thus, any rough set, in contrast to a crisp set, has a non-empty boundary region.

One can define the following four basic classes of rough sets, that is, four categories of vagueness:

$$\begin{aligned}
&X \text{ is } \textit{roughly } B\textit{-definable} \text{ iff } \mathsf{LOW}_B(X) \neq \varnothing \text{ and } \mathsf{UPP}_B(X) \neq U, \qquad (3.6)\\
&X \text{ is } \textit{internally } B\textit{-indefinable} \text{ iff } \mathsf{LOW}_B(X) = \varnothing \text{ and } \mathsf{UPP}_B(X) \neq U,\\
&X \text{ is } \textit{externally } B\textit{-indefinable} \text{ iff } \mathsf{LOW}_B(X) \neq \varnothing \text{ and } \mathsf{UPP}_B(X) = U,\\
&X \text{ is } \textit{totally } B\textit{-indefinable} \text{ iff } \mathsf{LOW}_B(X) = \varnothing \text{ and } \mathsf{UPP}_B(X) = U.
\end{aligned}$$

The intuitive meaning of this classification is the following.

If X is roughly B-*definable*, this means that we are able to decide for some elements of U that they belong to X and for some elements of U we are able to decide that they belong to $-X$, using B.

If X is internally B-indefinable, this means that we are able to decide about some elements of U that they belong to $-X$, but we are unable to decide for any element of U that it belongs to X, using B.

If X is externally B-indefinable, this means that we are able to decide for some elements of U that they belong to X, but we are unable to decide, for any element of U that it belongs to $-X$, using B.

If X is totally B-indefinable, we are unable to decide for any element of U whether it belongs to X or $-X$, using B.

Thus, a set is *rough* (imprecise) if it has nonempty boundary region; otherwise, the set is crisp (precise). This is exactly the idea of vagueness proposed by Frege.

Let us observe that the definition of rough sets refers to data (knowledge) and is *subjective*, in contrast to the definition of classical sets, which is in some sense an *objective* one.

A rough set can also be characterized numerically by the following coefficient

$$\alpha_B(X) = \frac{|\mathsf{LOW}_B(X)|}{|\mathsf{UPP}_B(X)|}, \tag{3.7}$$

called the *accuracy of approximation*, where X is a nonempty set and $|S|$ denotes the cardinality of set S. [5] Obviously $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$ then X is *crisp* with respect to B (X is *precise* with respect to B), and otherwise, if $\alpha_B(X) < 1$ then X is *rough* with respect to B (X is *vague* with respect to B). The accuracy of approximation can be used to measure the quality of approximation of decision classes on the universe U. One can use another measure of accuracy defined by $1 - \alpha_B(X)$ or by $1 - \frac{|\mathsf{BN}_B(X)|}{|U|}$. Some other measures of approximation accuracy are also used, *for example*, based on entropy or some more specific properties of boundary regions (see, *e.g.* [64, 288, 317]). The choice of a relevant accuracy of approximation depends on a particular data set. Observe that the accuracy of approximation of X can be tuned by B. Another approach to accuracy of approximation can be based on the Variable Precision Rough Set Model (VPRSM) [389].

In the next section, we discuss decision rules (constructed over a selected set B of features or a family of sets of features) that are used in inducing classification algorithms (classifiers) making it possible to classify to decision classes unseen objects. Parameters that are tuned in searching for a classifier with the high quality are its description size (defined using decision rules) and its quality of classification (measured by the number of misclassified objects on a given set of objects). By selecting a proper balance between the accuracy of classification and the description size we expect to find the classifier with the high quality of classification also on unseen objects. This approach is based on the minimum description length principle [267, 268, 318].

3.3.2 Decision Systems and Decision Rules

Sometimes, we distinguish in an information system $\mathbb{A} = (U, A)$ a partition of A into two disjoint classes $C, D \subseteq A$ of attributes called *condition* and *decision* (*action*) attributes, respectively. The tuple $\mathbb{A} = (U, C, D)$ is called a *decision system* (or *decison table*[6]).

[5] The cardinality of set S is also denoted by "card(S)" instead of $|S|$.

[6] More precisely, decision tables are representations of decision systems.

Let $V = \bigcup\{V_a \mid a \in C\} \bigcup \{V_d \mid d \in D\}$. Atomic formulae over $B \subseteq C \cup D$ and V are expressions $a = v$ called *descriptors (selectors) over B and V* where $a \in B$ and $v \in V_a$. The set of formulae over B and V denoted by $\mathcal{F}(B,V)$, is the least set containing all atomic formulae over B and V and closed under the propositional connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation).

By $\|\varphi\|_{\mathbb{A}}$, we denote the meaning of $\varphi \in \mathcal{F}(B,V)$ in the decision system $\mathbb{A}$ which is the set of all objects in U with the property φ. These sets are defined by $\|a = v\|_{\mathbb{A}} = \{x \in U \mid a(x) = v\}$, $\|\varphi \wedge \varphi'\|_{\mathbb{A}} = \|\varphi\|_{\mathbb{A}} \cap \|\varphi'\|_{\mathbb{A}}$; $\|\varphi \vee \varphi'\|_{\mathbb{A}} = \|\varphi\|_{\mathbb{A}} \cup \|\varphi'\|_{\mathbb{A}}$; $\|\neg\varphi\|_{\mathbb{A}} = U - \|\varphi\|_{\mathbb{A}}$. The formulae from $\mathcal{F}(C,V)$ and $\mathcal{F}(D,V)$ are called *condition formulae of* $\mathbb{A}$ and *decision formulae of* $\mathbb{A}$, respectively.

Any object $x \in U$ belongs to the *decision class* $\|\bigwedge_{d \in D} d = d(x)\|_{\mathbb{A}}$ of $\mathbb{A}$. All decision classes of $\mathbb{A}$ create a partition U/D of the universe U.

A *decision rule* for $\mathbb{A}$ is any expression of the form $\varphi \Rightarrow \psi$, where $\varphi \in \mathcal{F}(C,V)$, $\psi \in \mathcal{F}(D,V)$, and $\|\varphi\|_{\mathbb{A}} \neq \varnothing$. Formulae φ and ψ are referred to as the *predecessor* and the *successor* of decision rule $\varphi \Rightarrow \psi$. Decision rules are often called "$IF \ldots THEN \ldots$" rules. Such rules are used in machine learning (see, *e.g.* [97]).

Decision rule $\varphi \Rightarrow \psi$ is *true* in $\mathbb{A}$ if and only if $\|\varphi\|_{\mathbb{A}} \subseteq \|\psi\|_{\mathbb{A}}$. Otherwise, one can measure its *truth degree* by introducing some inclusion measure of $\|\varphi\|_{\mathbb{A}}$ in $\|\psi\|_{\mathbb{A}}$. Let us denote by $|\varphi|$ the number of objects from U that satisfies formula φ, that is, the cardinality of $\|\varphi\|_{\mathbb{A}}$. According to Łukasiewicz [142], one can assign to formula φ the value $\frac{|\varphi|}{|U|}$, and to the implication $\varphi \Rightarrow \psi$ the fractional value $\frac{|\varphi \wedge \psi|}{|\varphi|}$, under the assumption that $\|\varphi\| \neq \varnothing$. Proposed by Łukasiewicz, that fractional part was much later adapted by machine learning and data mining literature, *for example* in the definitions of the accuracy of decision rules or confidence of association rules.

Each object x of a decision system determines a *decision rule*

$$\bigwedge_{a \in C} a = a(x) \Rightarrow \bigwedge_{d \in D} d = d(x). \tag{3.8}$$

For any decision system $\mathbb{A} = (U,C,D)$ one can consider a *generalized decision* function $\partial_A : U \longrightarrow \mathcal{P}(INF(D))$ defined by

$$\partial_A(x) = \left\{ i \in INF(D) \ : \ \exists x' \in U \left[\, (x',x) \in IND_C \text{ and } Inf_D(x') = i \right] \right\}, \tag{3.9}$$

where $A = C \cup D$, $\mathcal{P}(INF(D))$ is the powerset of the set $INF(D)$ of all possible decision signatures.

The decision system $\mathbb{A}$ is called *consistent (deterministic)*, if $|\partial_A(x)| = 1$, for any $x \in U$. Otherwise, $\mathbb{A}$ is said to be *inconsistent (non-deterministic)*. Hence, decision system is inconsistent if it consists of some objects with different decisions but indiscernible with respect to condition attributes. Any set consisting of all objects with the same generalized decision value is called a *generalized decision class*.

Now, one can consider certain (possible) rules (see, *e.g.* [84, 89]) for decision classes defined by the lower (upper) approximations of such generalized decision

classes of $\mathbb{A}$. This approach can be extend, using the relationships of rough sets with the Dempster-Shafer theory (see, *e.g.* [278, 288]), by considering rules relative to decision classes defined by the lower approximations of unions of decision classes of $\mathbb{A}$.

Numerous methods have been developed for generation of different types of decision rules, and the reader can find by himself in the literature on rough sets. Usually, one is searching for decision rules (semi) optimal with respect to some optimization criteria describing quality of decision rules in concept approximations.

In the case of searching for concept approximation in an extension of a given universe of objects (sample), the following steps are typical. When a set of rules has been induced from a decision system containing a set of training examples, they can be inspected to see whether they reveal any novel relationships between attributes that are worth pursuing for further research. Furthermore, the rules can be applied to a set of unseen cases in order to estimate their classification power. For a systematic overview of rule application methods the reader is referred to the literature (see, *e.g.* [16, 151]).

3.3.3 Dependency of Attributes

Another important issue in data analysis is discovering dependencies between attributes in a given decision system $\mathbb{A} = (U, C, D)$. Intuitively, a set of attributes D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if the values of attributes from C uniquely determine the values of attributes from D. In other words, D depends totally on C, if there exists a functional dependency between values of C and D. Hence, $C \Rightarrow D$ if and only if the rule (3.8) is true on $\mathbb{A}$ for any $x \in U$. In general, D can depend partially on C. Formally, such a partial dependency can be defined in the following way.

We will say that D *depends on* C to a *degree* k $(0 \leq k \leq 1)$, denoted $C \Rightarrow_k D$, if

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|}, \tag{3.10}$$

where

$$POS_C(D) = \bigcup_{X \in U/D} \mathsf{LOW}_C(X), \tag{3.11}$$

called a *positive region* of the partition U/D with respect to C, is the set of all elements of U that can be uniquely classified to blocks of the partition U/D, by means of C.

If $k = 1$ we say that D *depends totally* on C, and if $k < 1$, we say that D *depends partially* (to *degree* k) on C. If $k = 0$ then the *positive region* of the partition U/D with respect to C is empty.

The coefficient k expresses the ratio of all elements of the universe, which can be properly classified to blocks of the partition U/D, employing attributes C and will be called the *degree of the dependency.*

It can be easily seen that if D depends totally on C then $IND_C \subseteq IND_D$. It means that the partition generated by C is finer than the partition generated by D. Notice, that the concept of dependency discussed above corresponds to that considered in relational databases.

Summing up: D is *totally* (*partially*) dependent on C, if *all* (*some*) elements of the universe U can be uniquely classified to blocks of the partition U/D, employing C.

Observe that (3.10) defines only one of possible measures of dependency between attributes (see, *e.g.* [316]). One also can compare the dependency discussed in this section with dependencies considered in databases [63].

3.3.4 Reduction of Attributes

We often face a question whether we can remove some data from a data-table preserving its basic properties, that is – whether a table contains some superfluous data.

Let us express this idea more precisely.

Let $C, D \subseteq A$ be sets of condition and decision attributes respectively. We will say that $C' \subseteq C$ is a *D-reduct* (reduct with *respect* to D) of C, if C' is a minimal subset of C such that

$$\gamma(C,D) = \gamma(C',D). \tag{3.12}$$

The intersection of all D-reducts is called a *D-core* (core with *respect* to D). Because the core is the intersection of all reducts, it is included in every reduct, that is, each element of the core belongs to some reduct. Thus, in a sense, the core is the most important subset of attributes, since none of its elements can be removed without affecting the classification power of attributes. Certainly, the geometry of reducts can be more compound. For example, the core can be empty but there can exist a partition of reducts into a few sets with non-empty intersection.

Many other kinds of reducts and their approximations are discussed in the literature (see, *e.g.* [20, 172, 175, 279, 314, 317, 318, 122, 169, 333]). For example, if one change the condition (3.12) to $\partial_A(x) = \partial_B(x)$, (where $A = C \cup D$ and $B = C' \cup D$) then the defined reducts are preserving the generalized decision. Other kinds of reducts are preserving, *for example*, (i) the distance between attribute value vectors for any two objects, if this distance is greater than a given threshold [279], (ii) the distance between entropy distributions between any two objects, if this distance exceeds a given threshold [314, 317] or (iii) the so-called reducts relative to object used for generation of decision rules [20]. There are some relationships between different kinds of reducts. If B is a reduct preserving the generalized decision, then in B is included a reduct preserving the positive region. For mentioned above reducts based

on distances and thresholds one can find analogous dependency between reducts relative to different thresholds. By choosing different kinds of reducts we select different degrees to which information encoded in data is preserved. Reducts are used for building data models. Choosing a particular reduct or a set of reducts has impact on the model size as well as on its quality in describing a given data set. The model size together and the model quality are two basic components tuned in selecting relevant data models. This is known as the minimum length principle (see, *e.g.* [267, 268, 317, 318]). Selection of relevant kinds of reducts is an important step in building data models. It turns out that the different kinds of reducts can be efficiently computed using heuristics based, *for example*, on the Boolean reasoning approach [31, 32, 30, 36].

3.3.5 Discernibility and Boolean Reasoning

Methodologies devoted to data mining, knowledge discovery, decision support, pattern classification and approximate reasoning require tools for discovering *templates (patterns)* in data and classifying them into certain *decision classes*. Templates are in many cases most frequent sequences of events, most probable events, regular configurations of objects, the decision rules of high quality, standard reasoning schemes. Tools for discovering and classifying of templates are based on *reasoning schemes* rooted in various paradigms [51]. Such patterns can be extracted from data by means of methods based, *for example*, on Boolean reasoning and discernibility.

The discernibility relations are closely related to indiscernibility and belong to the most important relations considered in rough set theory.

The ability to discern between perceived objects is important for constructing many entities like reducts, decision rules or decision algorithms. In the standard approach, the discernibility relation $\mathcal{DIS}_B \subseteq U \times U$ is defined by $x\ \mathcal{DIS}_B\ y$ if and only if $non(x\ \mathcal{IND}_B\ y)$, that is, $B(x) \cap B(y) = \varnothing$. However, this is, in general, not the case for generalized approximation spaces. For example, in the case of some of such spaces, for any object x may be given a family $F(x)$ with more than one elementary granules (neighbourhoods) such that $x \in \mathcal{I}(x)$ for any $\mathcal{I}(x) \in F(x)$. Then, one can define that objects x, y are discernible if and only if $\mathcal{I}(x) \cap \mathcal{I}(y) = \varnothing$ for some $\mathcal{I}(x) \in F(x)$ and $\mathcal{I}(y) \in F(y)$ and indiscernibility may be not the negation of this condition, *for example*, objects x, y are defined as indiscernible if and only if $\mathcal{I}(x) \cap \mathcal{I}(y) \neq \varnothing$ for some $\mathcal{I}(x) \in F(x)$ and $\mathcal{I}(y) \in F(y)$.

The idea of Boolean reasoning is based on construction for a given problem P of a corresponding Boolean function f_P with the following property: the solutions for the problem P can be decoded from prime implicants of the Boolean function f_P. Let us mention that to solve real-life problems it is necessary to deal with Boolean functions having large number of variables.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed for computing of many important ingredients for applications. These applications include generation of reducts and their approximations,

decision rules, association rules, discretization of real value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data as well as conflict resolution or negotiation (see, *e.g.* [20, 172, 175, 279, 314, 317, 318, 169]).

Most of the problems related to generation of the above mentioned entities are NP-complete or NP-hard. However, it was possible to develop efficient heuristics returning suboptimal solutions of the problems. The results of experiments on many data sets are very promising. They show very good quality of solutions generated by the heuristics in comparison with other methods reported in literature (*e.g.* with respect to the classification quality of unseen objects). Moreover they are very efficient from the point of view of time necessary for computing of the solution. Many of these methods are based on discernibility matrices. Note that it is possible to compute the necessary information about these matrices using directly[7] information or decision systems (*e.g.* sorted in preprocessing [16, 168, 178, 373]) that significantly improve the efficiency of algorithms.

It is important to note that the methodology makes it possible to construct heuristics having a very important *approximation property* that can be formulated as follows: expressions, called *approximate implicants*, generated by heuristics that are *close* to prime implicants define approximate solutions for the problem.

3.3.6 Rough Membership

Let us observe that rough sets can be also defined employing the rough membership function (see Eq. 3.13) instead of approximation [219]. That is, consider

$$\mu_X^B : U \to [0,1],$$

defined by

$$\mu_X^B(x) = \frac{|B(x) \cap X|}{|X|}, \tag{3.13}$$

where $x \in X \subseteq U$. The value $\mu_X^B(x)$ can be interpreted as the degree that x belongs to X in view of knowledge about x expressed by B or the degree to which the elementary granule $B(x)$ is included in the set X. This means that the definition reflects a subjective knowledge about elements of the universe, in contrast to the classical definition of a set.

The rough membership function can also be interpreted as the conditional probability that x belongs to X given B. This interpretation was used by several researchers in the rough set community (see, *e.g.* [87, 317, 357, 372, 393, 382], [389]). Note also that the ratio on the right-hand side of the equation (3.13) is known as the confidence coefficient in data mining [97, 115]. It is worthwhile to mention that set inclusion

[7] I.e. without the necessity of generation and storing of the discernibility matrices.

to a degree has been considered by Łukasiewicz [142] in studies on assigning fractional truth values to logical formulas.

On can observe that the rough membership function has the following properties [219]:

1) $\mu_X^B(x) = 1$ *iff* $x \in \mathsf{LOW}_B(X)$,
2) $\mu_X^B(x) = 0$ *iff* $x \in U - \mathsf{UPP}_B(X)$,
3) $0 < \mu_X^B(x) < 1$ *iff* $x \in \mathsf{BN}_B(X)$,
4) $\mu_{U-X}^B(x) = 1 - \mu_X^B(x)$ *for any* $x \in U$,
5) $\mu_{X\cup Y}^B(x) \geq \max\left(\mu_X^B(x),\ \mu_Y^B(x)\right)$ *for any* $x \in U$,
6) $\mu_{X\cap Y}^B(x) \leq \min\left(\mu_X^B(x),\ \mu_Y^B(x)\right)$ *for any* $x \in U$.

From the properties it follows that the rough membership differs essentially from the fuzzy membership [385], for properties 5) and 6) show that the membership for union and intersection of sets, in general, cannot be computed – as in the case of fuzzy sets – from their constituents membership. Thus formally the rough membership is different from fuzzy membership. Moreover, the rough membership function depends on an available knowledge (represented by attributes from B). Besides, the rough membership function, in contrast to fuzzy membership function, has a probabilistic flavor.

Let us also mention that rough set theory, in contrast to fuzzy set theory, clearly distinguishes two very important concepts, vagueness and uncertainty, very often confused in the AI literature. Vagueness is the property of concepts. Vague concepts can be approximated using the rough set approach [287]. Uncertainty is the property of elements of a set or a set itself (*e.g.* only examples and/or counterexamples of elements of a considered set are given). Uncertainty of elements of a set can be expressed by the rough membership function.

Both fuzzy and rough set theory represent two different approaches to vagueness. Fuzzy set theory addresses *gradualness* of knowledge, expressed by the fuzzy membership, whereas rough set theory addresses *granularity* of knowledge, expressed by the indiscernibility relation. A nice illustration of this difference has been given by Dider Dubois and Henri Prade [49] in the following example. In image processing fuzzy set theory refers to gradualness of gray level, whereas rough set theory is about the size of pixels.

Consequently, both theories are not competing but are rather complementary. In particular, the rough set approach provides tools for approximate construction of fuzzy membership functions. The rough-fuzzy hybridization approach proved to be successful in many applications (see, *e.g.* [196, 200]).

Interesting discussion of fuzzy and rough set theory in the approach to vagueness can be found in [266]. Let us also observe that fuzzy set and rough set theory are not a remedy for classical set theory difficulties.

One of the consequences of perceiving objects by information about them is that for some objects one cannot decide whether they belong to a given set or not. However, one can estimate the degree to which objects belong to sets. This is a crucial observation in building foundations for approximate reasoning. Dealing with imperfect

knowledge implies that one can only characterize satisfiability of relations between objects to a degree, not precisely. One of the fundamental relations on objects is a rough inclusion relation describing that objects are parts of other objects to a degree. The rough mereological approach [248, 199, 249, 250, 252] based on such a relation is an extension of the Leśniewski mereology [128].

3.4 Generalizations of Approximation Spaces

The rough set concept can be defined quite generally by means of topological operations, *interior* and *closure*, called *approximations* [245]. It was observed in [212] that the key to the presented approach is provided by the exact mathematical formulation of the concept of approximative (rough) equality of sets in a given approximation space. In [215], an approximation space is represented by the pair $(U, \mathcal{R})$, where U is a universe of objects, and $\mathcal{R} \subseteq U \times U$ is an indiscernibility relation defined by an attribute set (i.e. $\mathcal{R} = IND_A$ for some attribute set A). In this case, $\mathcal{R}$ is the equivalence relation. Let $[x]_{\mathcal{R}}$ denote an equivalence class of an element $x \in U$ under the indiscernibility relation $\mathcal{R}$, where $[x]_{\mathcal{R}} = \{y \in U : x\mathcal{R}y\}$.

In this context, $\mathcal{R}$-approximations of any set $X \subseteq U$ are based on the exact (crisp) containment of sets. Then set approximations are defined as follows:

- $x \in U$ belongs with certainty to $X \subseteq U$ (i.e. x belongs to the $\mathcal{R}$-lower approximation of X), if $[x]_{\mathcal{R}} \subseteq X$.
- $x \in U$ possibly belongs $X \subseteq U$ (i.e. x belongs to the $\mathcal{R}$-upper approximation of X), if $[x]_{\mathcal{R}} \cap X \neq \emptyset$.
- $x \in U$ belongs with certainty neither to the X nor to $U - X$ (i.e. x belongs to the $\mathcal{R}$-boundary region of X), if $[x]_{\mathcal{R}} \cap (U - X) \neq \emptyset$ and $[x]_{\mathcal{R}} \cap X \neq \emptyset$.

Our knowledge about the approximated concepts is often partial and uncertain [83]. For example, concept approximation should be constructed from examples and counterexamples of objects for the concepts [97]. Hence, concept approximations constructed from a given sample of objects are extended, using inductive reasoning, on objects not yet observed.

Several generalizations of the classical rough set approach based on approximation spaces defined as pairs of the form $(U, \mathcal{R})$, where $\mathcal{R}$ is the equivalence relation (called indiscernibility relation) on the set U have been reported in the literature (see, *e.g.* [132, 134, 251, 284, 356, 379, 380, 381, 383, 308, 302, 331, 301, 303]) [8].

Let us mention two of them.

The concept approximations should be constructed under dynamically changing environments [287, 307]. This leads to a more complex situation where the boundary regions are not crisp sets, which is consistent with the postulate of the higher order

[8] Among extensions not discussed in this chapter is the rough set approach to multicriteria decision-making (see *e.g.* [75, 76, 77, 78, 82, 231, 243, 325, 79, 120, 80, 81, 74, 323] and also the Chapter *jMAF - Dominance-based Rough Set Data Analysis Framework* by J. Błaszczyński, S. Greco, B. Matarazzo, R. Słowiński, M. Szeląg in this book (chapter 5)).

vagueness considered by philosophers (see, *e.g.* [111]). Different aspects of vagueness in the rough set framework are discussed, *for example*, in [145, 184, 186, 266, 287]. It is worthwhile to mention that a rough set approach to the approximation of compound concepts has been developed. For such concepts, it is hardly possible to expect that they can be approximated with the high quality by the traditional methods [35, 364]. The approach is based on hierarchical learning and ontology approximation [13, 24, 177, 199, 294]. Approximation of concepts in distributed environments is discussed in [285]. A survey of algorithmic methods for concept approximation based on rough sets and Boolean reasoning in presented, *for example*, in [16, 281, 169, 13].

A generalized approximation space[9] can be defined by a tuple $\mathbb{AS} = (U, \mathcal{I}, \nu)$ where $\mathcal{I}$ is the *uncertainty function* defined on U with values in the powerset $\mathcal{P}(U)$ of U ($\mathcal{I}(x)$ is the *neighboorhood* of x) and ν is the *inclusion function* defined on the Cartesian product $\mathcal{P}(U) \times \mathcal{P}(U)$ with values in the interval $[0,1]$ measuring the degree of inclusion of sets [297]. The lower and upper approximation operations can be defined in $\mathbb{AS}$ by

$$\mathsf{LOW}_{\mathbb{AS}}(X) = \{x \in U : \nu(\mathcal{I}(x), X) = 1\}, \tag{3.14}$$

$$\mathsf{UPP}_{\mathbb{AS}}(X) = \{x \in U : \nu(\mathcal{I}(x), X) > 0\}. \tag{3.15}$$

In the standard case, $\mathcal{I}(x)$ is equal to the equivalence class $B(x)$ of the indiscernibility relation IND_B; in case of tolerance (similarity) relation $\tau \subseteq U \times U$ [256] we take $\mathcal{I}(x) = [x]_\tau = \{y \in U : x \,\tau\, y\}$, that is, $\mathcal{I}(x)$ is equal to the tolerance class of τ defined by x. The standard rough inclusion relation ν_{SRI} is defined for $X, Y \subseteq U$ by

$$\nu_{SRI}(X,Y) = \begin{cases} \dfrac{|X \cap Y|}{|X|}, & \text{if } X \text{ is non} - \text{empty}, \\ 1, & \text{otherwise.} \end{cases} \tag{3.16}$$

For applications, it is important to have some constructive definitions of $\mathcal{I}$ and ν.

One can consider another way to define $\mathcal{I}(x)$. Usually together with $\mathbb{AS}$ we consider some set $\mathcal{F}$ of formulae describing sets of objects in the universe U of $\mathbb{AS}$, defined by semantics $\|\cdot\|_{\mathbb{AS}}$, that is, $\|\alpha\|_{\mathbb{AS}} \subseteq U$ for any $\alpha \in \mathcal{F}$[10]. Now, one can take the set

$$N_{\mathcal{F}}(x) = \{\alpha \in \mathcal{F} : x \in \|\alpha\|_{\mathbb{AS}}\}, \tag{3.17}$$

and $\mathcal{I}_\circ(x) = \{\|\alpha\|_{\mathbb{AS}} : \alpha \in N_{\mathcal{F}}(x)\}$. Hence, more general uncertainty functions having values in $\mathcal{P}(\mathcal{P}(U))$ can be defined and in the consequence different definitions of approximations are considered. For example, one can consider the following definitions of approximation operations in this approximation space $\mathbb{AS}_\circ = (U, \mathcal{I}_\circ, \nu)$:

$$\mathsf{LOW}_{\mathbb{AS}_\circ}(X) = \{x \in U : \nu(Y, X) = 1 \text{ for some } Y \in \mathcal{I}(x)\}, \tag{3.18}$$

$$\mathsf{UPP}_{\mathbb{AS}_\circ}(X) = \{x \in U : \nu(Y, X) > 0 \text{ for any } Y \in \mathcal{I}(x)\}. \tag{3.19}$$

[9] More general cases are considered, *for example*, in [301, 303].

[10] If $\mathbb{AS} = (U, \mathcal{I}, \nu)$ then we will also write $\|\alpha\|_U$ instead of $\|\alpha\|_{\mathbb{AS}}$.

There are also different forms of rough inclusion functions. Let us consider two examples.

In the first example of a rough inclusion function, a threshold $t \in (0, 0.5)$ is used to relax the degree of inclusion of sets. The rough inclusion function ν_t is defined by

$$\nu_t(X,Y) = \begin{cases} 1 & \text{if } \nu_{SRI}(X,Y) \geq 1-t, \\ \frac{\nu_{SRI}(X,Y)-t}{1-2t} & \text{if } t \leq \nu_{SRI}(X,Y) < 1-t, \\ 0 & \text{if } \nu_{SRI}(X,Y) \leq t. \end{cases} \tag{3.20}$$

This is an interesting "rough-fuzzy" example because we put the standard rough membership function as an argument into the formula often used for fuzzy membership functions.

One can obtain approximations considered in the variable precision rough set approach (VPRSM) [389] by substituting in (3.14)-(3.15) the rough inclusion function ν_t defined by (3.20) instead of ν, assuming that Y is a decision class and $\mathcal{I}(x) = B(x)$ for any object x, where B is a given set of attributes.

Another example of application of the standard inclusion was developed by using probabilistic decision functions. For more detail the reader is referred to [303, 349, 316, 317].

The rough inclusion relation can be also used for function approximation [308, 301, 303] and relation approximation [329]. In the case of function approximation the inclusion function ν^* for subsets $X, Y \subseteq U \times U$, where $U \subseteq \mathbb{R}$ and $\mathbb{R}$ is the set of real numbers, is defined by

$$\nu^*(X,Y) = \begin{cases} \dfrac{|\pi_1(X \cap Y)|}{|\pi_1(X)|} & \text{if } \pi_1(X) \neq \varnothing, \\ 1 & \text{if } \pi_1(X) = \varnothing, \end{cases} \tag{3.21}$$

where π_1 is the projection operation on the first coordinate. Assume now, that X is a cube, and Y is the graph $G(f)$ of the function $f : \mathbb{R} \longrightarrow \mathbb{R}$. Then, *for example*, X is in the lower approximation of f if the projection on the first coordinate of the intersection $X \cap G(f)$ is equal to the projection of X on the first coordinate. This means that the part of the graph $G(f)$ is "well" included in the box X, that is, for all arguments that belong to the box projection on the first coordinate the value of f is included in the box X projection on the second coordinate. This approach was extended in several papers (see, *e.g.* [349, 303]).

The approach based on inclusion functions has been generalized to the *rough mereological approach* [199, 250, 249, 252]. The inclusion relation $x\mu_r y$ with the intended meaning *x is a part of y to a degree at least r* has been taken as the basic notion of the rough mereology being a generalization of the Leśniewski mereology [128, 129]. Research on rough mereology has shown importance of another notion, namely *closeness* of compound objects (*e.g.* concepts). This can be defined by $x\, cl_{r,r'}\, y$ if and only if $x\, \mu_r\, y$ and $y\, \mu_{r'}\, x$.

Rough mereology offers a methodology for synthesis and analysis of objects in a distributed environment of intelligent agents, in particular, for synthesis of objects satisfying a given specification to a satisfactory degree or for control in such a complex environment. Moreover, rough mereology has been used for developing the foundations of the *information granule calculi*, aiming at formalization of the Computing with Words paradigm, formulated by Lotfi Zadeh [386]. More complex information granules are defined recursively using already defined information granules and their measures of inclusion and closeness. Information granules can have complex structures like classifiers or approximation spaces. Computations on information granules are performed to discover relevant information granules, *for example*, patterns or approximation spaces for compound concept approximations.

Usually, there are considered families of approximation spaces labelled by some parameters. By tuning such parameters according to chosen criteria (*e.g.* minimal description length) one can search for the optimal approximation space for concept description (see, *e.g.* [16, 169, 13]).

3.5 Rough Sets and Induction

Granular formulas are constructed from atomic formulas corresponding to the considered attributes (see, *e.g.* [222, 221, 301, 303]). In the consequence, the satisfiability of such formulas is defined if the satisfiability of atomic formulas is given as the result of sensor measurement. Let us consider two information systems $\mathbb{A} = (U, C, D)$ and its extension $\mathbb{A}^* = (U^*, C)$ having the same set of attributes C (more precisely, the set o attributes in $\mathbb{A}$ is obtained by restricting to U attributes from $\mathbb{A}^*$ defined on $U^* \supseteq U$).

Hence, one can consider for any constructed formula α over atomic formulas its semantics $\|\alpha\|_{\mathbb{A}} \subseteq U$ over U as well as the semantics $\|\alpha\|_{\mathbb{A}^*} \subseteq U^*$ over U^* (see Figure 3.2).

The difference between these two cases is the following. In the case of U, one can compute $\|\alpha\|_{\mathbb{A}} \subseteq U$, but in the case $\|\alpha\|_{\mathbb{A}^*} \subseteq U^*$, for any object from $U^* - U$, there is no information about its membership relative to $\|\alpha\|_{\mathbb{A}^*} - \|\alpha\|_{\mathbb{A}}$. One can estimate the satisfiability of α for objects $u \in U^* - U$ only after some relevant sensory measurements on u are performed. In particular, one can use some methods for estimation of relationships among semantics of formulas over U^* using the relationships among semantics of these formulas over U. For example, one can apply statistical methods. This step is crucial in investigation of extensions of approximation spaces relevant for inducing classifiers from data.

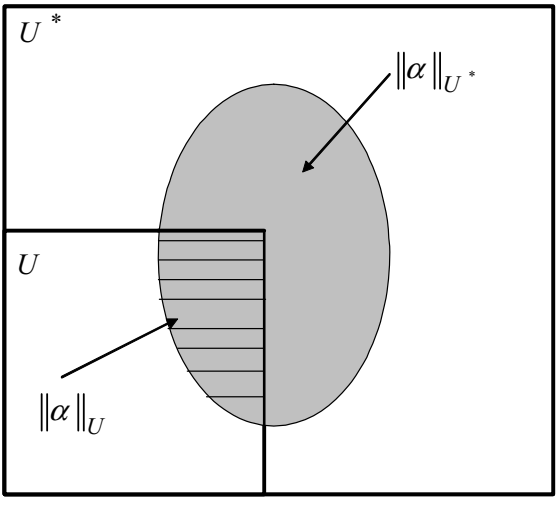


Fig. 3.2. Two semantics of α over U and U^*, respectively

3.5.1 Rough Sets and Classifiers

In this section, we consider the problem of approximation of concepts over a universe U^∞ (concepts that are subsets of U^∞). We assume that the concepts are perceived only through some subsets of U^∞ called samples. This is a typical situation in the machine learning, pattern recognition or data mining approaches [97, 115, 135]. We explain the rough set approach to induction of concept approximations using the generalized approximation spaces of the form $\mathbb{AS} = (U, I, \nu)$ defined in Section 3.4.

Let $U \subseteq U^\infty$ be a finite sample. By Π_U we denote a perception function from $\mathcal{P}(U^\infty)$ into $\mathcal{P}(U)$ defined by $\Pi_U(C) = C \cap U$ for any concept $C \subseteq U^\infty$.

Let us consider first an illustrative example.

We assume that there is given an information system $\mathbb{A} = (U, A)$ and let us assume that for some $C \subseteq U^\infty$ there is given the set $\Pi_U(C) = C \cap U$. In this way we obtain a decision system $\mathbb{AS}_d = (U, A, d)$, where $d(x) = 1$ if $x \in C \cap U$ and $d(x) = 0$, otherwise.

We would like to illustrate how from the decision function d may be induced a decision function d^* defined over U^∞ which can be treated as an approximation of the characteristic function of C.

Let us assume that $RULES(\mathbb{AS}_d)$ is a set of decision rules induced by some rule generation method from $\mathbb{AS}_d$. For any object $x \in U^\infty$, let $MatchRules(\mathbb{AS}_d, x)$ be the set of rules from $RULES(\mathbb{AS}_d)$ supported by x (see, *e.g.* [20]).

Let $C_1 = C$ and $C_0 = U^\infty \setminus C$. Now, for $k = 1, 0$ one can define the rough membership functions $\mu_k : U^\infty \to [0, 1]$ in the following way:

1. Let $R_k(x)$ be the set of all decision rules from $MatchRules(\mathbb{AS}_d, x)$ for C_k, that is, decision rules from $MatchRules(\mathbb{AS}_d, x)$ with right hand side $d = k$
2. We define real values $w_k(x)$, where $w_1(x)$ is called the weight "for" and $w_0(x)$ the weight "against" membership of the object x in C, respectively, by

$$w_k(x) = \sum_{r \in R_k(x)} strength(r),$$

   where $strength(r)$ is a normalized function depending on *length*, *support*, $confidence$ of the decision rule r and on some global information about the decision table $\mathbb{AS}_d$ such as the table size or the class distribution (see [20]).
3. Finally, one can define the value of $\mu_k(x)$ by

$$\mu_k(x) = \begin{cases} \text{undefined} & \text{if } \max(w_k(x), w_{1-k}(x)) < \omega \\ 0 & \text{if } w_{1-k}(x) - w_k(x) \geq \theta \text{ and } w_{1-k}(x) > \omega \\ 1 & \text{if } w_k(x) - w_{1-k}(x) \geq \theta \text{ and } w_k(x) > \omega \\ \frac{\theta + (w_k(x) - w_{1-k}(x))}{2\theta} & \text{in other cases,} \end{cases}$$

   where ω, θ are parameters set by user.

Now, for computing of the value $d^*(x)$ for $x \in U^\infty$ the user should select a strategy resolving conflicts between values $\mu_1(x)$ and $\mu_0(x)$ representing, in a sense votes "for" and "against" membership of x in C, respectively. Note that for some cases x due to the small differences between these values the selected strategy may not produce the definite answer and these cases will create the boundary region.

Let us consider a generalized approximation space $\mathbb{AS} = (U, I, \nu_{SRI})$, where $I(x) = A(x)$ for $x \in U$. Now, we would like to check how this approximation space may be inductively extended so that in the induced approximation space we may define approximation of the concept C or in other words the approximation of the decision function d^*.

Hence, the problem we are considering is how to extend the approximations of $\Pi_U(C)$ defined by $\mathbb{AS}$ to an approximation of C over U^∞. We show that the problem can be described as searching for an extension $\mathbb{AS}_C = (U^\infty, \mathcal{I}_C, \nu_C)$ of the approximation space $\mathbb{AS}$, relevant for approximation of C. This requires to show how to extend the inclusion function ν from subsets of U to subsets of U^∞ that are relevant for the approximation of C. Observe that for the approximation of C, it is enough to induce the necessary values of the inclusion function ν_C without knowing the exact value of $\mathcal{I}_C(x) \subseteq U^\infty$ for $x \in U^\infty$.

Let $\mathbb{AS}$ be a given approximation space for $\Pi_U(C)$ and let us consider a language L in which the neighbourhood $\mathcal{I}(x) \subseteq U$ is expressible by a formula $pat(x)$, for any $x \in U$. It means that $\mathcal{I}(x) = \|pat(x)\|_U \subseteq U$ where $\|pat(x)\|_U$ denotes the meaning of $pat(x)$ restricted to the sample U. In case of rule-based classifiers, patterns of the form $pat(x)$ are defined by feature value vectors.

We assume that for any new object $x \in U^\infty - U$, we can obtain (*e.g.* as a result of sensor measurement) a pattern $pat(x) \in L$ with semantics $\|pat(x)\|_{U^\infty} \subseteq U^\infty$. However, the relationships between information granules over U^∞ like sets: $\|pat(x)\|_{U^\infty}$

and $\|pat(y)\|_{U^\infty}$, for different $x, y \in U^\infty$, are, in general, known only if they can be expressed by relationships between the restrictions of these sets to the sample U, that is, between sets $\Pi_U(\|pat(x)\|_{U^\infty})$ and $\Pi_U(\|pat(y)\|_{U^\infty})$.

The set of patterns $\{pat(x) : x \in U\}$ is usually not relevant for approximation of the concept $C \subseteq U^\infty$. Such patterns are too specific or not enough general and can directly be applied only to a very limited number of new objects. However, by using some generalization strategies, one can search, in a family of patterns definable from $\{pat(x) : x \in U\}$ in L, for such new patterns that are relevant for approximation of concepts over U^∞. Let us consider a subset $PATTERNS(\mathbb{AS}, L, C) \subseteq L$ chosen as a set of pattern candidates for relevant approximation of a given concept C. For example, in case of rule based classifier one can search for such candidate patterns among sets definable by subsequences of feature value vectors corresponding to objects from the sample U. The set $PATTERNS(\mathbb{AS}, L, C)$ can be selected by using some quality measures checked on meanings (semantics) of its elements restricted to the sample U (like the number of examples from the concept $\Pi_U(C)$ and its complement that support a given pattern). Then, on the basis of properties of sets definable by these patterns over U we induce approximate values of the inclusion function ν_C on subsets of U^∞ definable by any of such pattern and the concept C.

Next, we induce the value of ν_C on pairs (X, Y) where $X \subseteq U^\infty$ is definable by a pattern from $\{pat(x) : x \in U^\infty\}$ and $Y \subseteq U^\infty$ is definable by a pattern from $PATTERNS(\mathbb{AS}, L, C)$.

Finally, for any object $x \in U^\infty - U$ we induce the approximation of the degree

$$\nu_C(\|pat(x)\|_{U^\infty}, C)$$

applying a conflict resolution strategy $Conflict\_res$ (a voting strategy, in case of rule based classifiers) to two families of degrees:

$$\{\nu_C(\|pat(x)\|_{U^\infty}, \|pat\|_{U^\infty}) : pat \in PATTERNS(\mathbb{AS}, L, C)\}, \tag{3.22}$$

$$\{\nu_C(\|pat\|_{U^\infty}, C) : pat \in PATTERNS(\mathbb{AS}, L, C)\}. \tag{3.23}$$

Values of the inclusion function for the remaining subsets of U^∞ can be chosen in any way – they do not have any impact on the approximations of C. Moreover, observe that for the approximation of C we do not need to know the exact values of uncertainty function I_C – it is enough to induce the values of the inclusion function ν_C. Observe that the defined extension ν_C of ν to some subsets of U^∞ makes it possible to define an approximation of the concept C in a new approximation space $\mathbb{AS}_C$.

Observe that one can also follow principles of Bayesian reasoning and use degrees of ν_C to approximate C (see, *e.g.* [217, 319, 322]).

Let us present yet another example of (inductive) extension $\mathbb{AS}^*$ of approximation space $\mathbb{AS}$ in the case of rule based classifiers. For details the reader is referred to, *for example*, [301, 303].

Let $h : [0,1] \to \{0, 1/2, 1\}$ be a function defined by

$$h(t) = \begin{cases} 1, \text{ if } t > \frac{1}{2}, \\ \frac{1}{2}, \text{ if } t = \frac{1}{2}, \\ 0, \text{ if } t < \frac{1}{2}. \end{cases} \tag{3.24}$$

We start with an extension of the uncertainty function and the rough inclusion function from U to U^*, where $U \subseteq U^*$:

$$\mathcal{I}(x) = \{\|lh(r)\|_{U^*} : x \in \|lh(r)\|_{U^*} \text{ and } r \in \textit{Rule\_set}\}, \tag{3.25}$$

where $x \in U^*$ and $lh(r)$ denotes the formula on the left-hand side of the rule r, and *Rule_set* is a set of decision rules induced from a given decision system $DT = (U, A, d)$. In this approach, the rough inclusion function is defined by

$$\nu_U(X,Z) = h\left(\frac{|\{Y \in X : Y \cap U \subseteq Z\}|}{|\{Y \in X : Y \cap U \subseteq Z\}| + |\{Y \in X : Y \cap U \subseteq U^* - Z\}|}\right), \tag{3.26}$$

where $X \subseteq \mathcal{P}(U^*)$, $X \neq \varnothing$ and $Z \subseteq U^*$. In case $X = \varnothing$ we set $\nu_U(\varnothing, Z) = 0$

The induced uncertainty and rough inclusion functions can now be used to define the lower approximation $\mathsf{LOW}_{\mathbb{AS}^*}(Z)$, the upper approximation $\mathsf{UPP}_{\mathbb{AS}^*}(Z)$, and the boundary region $\mathsf{BN}_{\mathbb{AS}^*}(Z)$ of $Z \subseteq U^*$ by:

$$\mathsf{LOW}_{\mathbb{AS}^*}(Z) = \{x \in U^* : \nu_U(\mathcal{I}(x), Z) = 1\}, \tag{3.27}$$

$$\mathsf{UPP}_{\mathbb{AS}^*}(Z) = \{x \in U^* : \nu_U(\mathcal{I}(x), Z) > 0\}, \tag{3.28}$$

and

$$\mathsf{BN}_{\mathbb{AS}^*}(Z) = \mathsf{UPP}_{\mathbb{AS}^*}(Z) - \mathsf{LOW}_{\mathbb{AS}^*}(Z). \tag{3.29}$$

In the example, we classify objects from U^* to the lower approximation of Z if majority of rules matching this object are voting for Z and to the upper approximation of Z if at least half of the rules matching x are voting for Z. Certainly, one can follow many other voting schemes developed in machine learning or by introducing less crisp conditions in the boundary region definition. The defined approximations can be treated as estimations of the exact approximations of subsets of U^* because they are induced on the basis of samples of such sets restricted to U only. One can use some standard quality measures developed in machine learning to calculate the quality of such approximations assuming that after estimation of approximations on U^* full information about membership for objects relative to the approximated subsets of U^* is uncovered analogously to the testing sets in machine learning.

In an analogous way, one can describe other class of classifiers used in machine learning and data mining such as neural networks or k-nn classifiers.

In this way, the rough set approach to induction of concept approximations can be explained as a process of inducing a relevant approximation space.

In [303] is presented the rough set approach to approximation of partially defined concepts (see also, *e.g.* [22, 23, 26, 302, 177, 169, 13, 331, 311, 301, 312]). The problems discussed in this chapter are crucial for building computer systems that assist researchers in scientific discoveries in many areas. Our considerations can be treated as a step towards foundations for modelling of granular computations inside of system that is based on granules called approximation spaces. Approximation spaces are fundamental granules used in searching for relevant complex granules called as data models, *for example*, approximations of complex concepts, functions or relations. The approach requires some generalizations of the approximation space concept introduced in [296, 297]. There are presented examples of rough set-based strategies for the extension of approximation spaces from samples of objects onto a whole universe of objects. This makes it possible to present foundations for inducing data models such as approximations of concepts or classifications analogous to the approaches for inducing different types of classifiers known in machine learning and data mining. Searching for relevant approximation spaces and data models are formulated as complex optimization problems. This optimization is performed relative to some measures that are some versions of the minimum length principle (MLP) [267, 268].

3.5.2 Inducing Relevant Approximation Spaces

A key task in granular computing is the information granulation process that leads to the formation of information aggregates (with inherent patterns) from a set of available objects. A methodological and algorithmic issue is the formation of transparent (understandable) information granules inasmuch as they should provide a clear and understandable description of patterns present in sample objects [9, 223]. Such a fundamental property can be formalized by a set of constraints that must be satisfied during the information granulation process. For example, in case of inducing granules such as classifiers, the constraints specify requirements for the quality of classifiers. Then, inducing of classifiers can be understood as searching for relevant approximation spaces (which can be treated as a spacial type of granules) relative to some properly selected optimization measures[11]. The selection of these optimization measures is not an easy task because they should guarantee that the (semi-) optimal approximation spaces selected relative to these criteria should allow us to construct classifiers of the high quality.

Let us consider some examples of optimization measures [170]. For example, the quality of an approximation space can be measured by:

$$Quality_1 : \mathcal{SAS}(U) \times \mathcal{P}(U) \to [0,1], \tag{3.30}$$

[11] Note that while there is a large literature on the covering based rough set approach (see, *e.g.* [388, 91]) still much more work should be done on (scalable) algorithmic searching methods for relevant approximation spaces in huge families of approximation spaces defined by many parameters determining neighbourhoods, inclusion measures and approximation operators.

where U is a non-empty set of objects and $\mathcal{SAS}(U)$ is a set of possible approximation spaces with the universe U.

Example 3.1. If $\mathsf{UPP}_{\mathbb{AS}}(X) \neq \varnothing$ for $\mathbb{AS} \in \mathcal{SAS}(U)$ and $X \subseteq U$ then

$$Quality_1(\mathbb{AS}, X) = \nu_{SRI}(\mathsf{UPP}_{\mathbb{AS}}(X), \mathsf{LOW}_{\mathbb{AS}}(X)) = \frac{|\mathsf{LOW}_{\mathbb{AS}}(X)|}{|\mathsf{UPP}_{\mathbb{AS}}(X)|}. \tag{3.31}$$

The value $1 - Quality_1(\mathbb{AS}, X)$ expresses the degree of completeness of our knowledge about X, given the approximation space $\mathbb{AS}$.

Example 3.2. In applications, we usually use another quality measure analogous to the minimum length principle [268, 268] where also the description length of approximation is included. Let us denote by $description(\mathbb{AS}, X)$ the description length of approximation of X in $\mathbb{AS}$. The description length may be measured, *for example*, by the sum of description lengths of algorithms testing membership for neighborhoods used in construction of the lower approximation, the upper approximation, and the boundary region of the set X. Then the quality $Quality_2(\mathbb{AS}, X)$ can be defined by

$$Quality_2(\mathbb{AS}, X) = g(Quality_1(\mathbb{AS}, X), description(\mathbb{AS}, X)), \tag{3.32}$$

where g is a relevant function used for fusion of values $Quality_1(\mathbb{AS}, X)$ and $description(\mathbb{AS}, X)$. This function g can reflect weights given by experts relative to both criteria.

One can consider different optimization problems relative to a given class $Set\_\mathbb{AS}$ of approximation spaces. For example, for a given $X \subseteq U$ and a threshold $t \in [0, 1]$, one can search for an approximation space $\mathbb{AS}$ satisfying the constraint $Quality_2(\mathbb{AS}, X) \geq t$.

Another example can be related to searching for an approximation space satisfying additionally the constraint $Cost(\mathbb{AS}) < c$ where $Cost(\mathbb{AS})$ denotes the cost of approximation space $\mathbb{AS}$ (e.g. measured by the number of attributes used to define neighbourhoods in $\mathbb{AS}$), and c is a given threshold. In the following example, we consider also costs of searching for relevant approximation spaces in a given family defined by a parameterized approximation space. Any parameterized approximation space $\mathbb{AS}_{\#,\$} = (U, I_\#, \nu_\$)$ is a family of approximation spaces. The cost of searching in such a family for a relevant approximation space for a given concept X approximation can be treated as a factor of the quality measure of approximation of X in $\mathbb{AS}_{\#,\$} = (U, I_\#, \nu_\$)$. Hence, such a quality measure of approximation of X in $\mathbb{AS}_{\#,\$}$ can be defined by

$$Quality_3(\mathbb{AS}_{\#,\$}, X) = h(Quality_2(\mathbb{AS}, X), Cost\_Search(\mathbb{AS}_{\#,\$}, X)), \tag{3.33}$$

where $\mathbb{AS}$ is the result of searching in $\mathbb{AS}_{\#,\$}$, $Cost\_Search(\mathbb{AS}_{\#,\$}, X)$ is the cost of searching in $\mathbb{AS}_{\#,\$}$ for $\mathbb{AS}$, and h is a fusion function, *for example*, assuming that the values of $Quality_2(\mathbb{AS}, X)$ and $Cost\_Search(\mathbb{AS}_{\#,\$}, X)$ are normalized to interval $[0, 1]$ h could be defined by a linear combination of $Quality_2(\mathbb{AS}, X)$ and $Cost\_Search(\mathbb{AS}_{\#,\$}, X)$ of the form

$$\lambda Quality_2(\mathbb{AS},X) + (1-\lambda)Cost\_Search(\mathbb{AS}_{\#,\$},X), \tag{3.34}$$

where $0 \leq \lambda \leq 1$ is a weight measuring an importance of quality and cost in their fusion.

We assume that the fusion functions g,h in the definitions of quality are monotonic relative to each argument.

Let $\mathbb{AS} \in Set\_\mathbb{AS}$ be an approximation space relevant for approximation of $X \subseteq U$, *that is*, $\mathbb{AS}$ is the optimal (or semi-optimal) relative to $Quality_2$. By

$$Granulation(\mathbb{AS}_{\#,\$}),$$

we denote a new parameterized approximation space obtained by granulation of $\mathbb{AS}_{\#,\$}$. For example, $Granulation(\mathbb{AS}_{\#,\$})$ can be obtained by reducing the number of attributes or inclusion degrees (*i.e.* possible values of the inclusion function). Let $\mathbb{AS}'$ be an approximation space in $Granulation(\mathbb{AS}_{\#,\$})$ obtained as the result of searching for optimal (semi-optimal) approximation space in $Granulation(\mathbb{AS}_{\#,\$})$ for approximation of X.

We assume that three conditions are satisfied:

- after granulation of $\mathbb{AS}_{\#,\$}$ to $Granulation(\mathbb{AS}_{\#,\$})$ the following property holds: the cost
$$Cost\_Search(Granulation(\mathbb{AS}_{\#,\$}),X), \tag{3.35}$$
is much lower than the cost $Cost\_Search(\mathbb{AS}_{\#,\$},X)$;
- $description(\mathbb{AS}',X)$ is much shorter than $description(\mathbb{AS},X)$, *that is*, the description length of X in the approximation space $\mathbb{AS}'$ is much shorter than the description length of X in the approximation space $\mathbb{AS}$;
- $Quality_1(\mathbb{AS},X)$ and $Quality_1(\mathbb{AS}',X)$ are sufficiently close.

The last two conditions should guarantee that the values

$$Quality_2(\mathbb{AS},X) \text{ and } Quality_2(\mathbb{AS}',X)$$

are comparable and this condition together with the first condition about the cost of searching should assure that

$$Quality_3(Granulation(\mathbb{AS}_{\#,\$},X)) \text{ is much better than } Quality_3(\mathbb{AS}_{\#,\$},X). \tag{3.36}$$

Certainly, the phrases already mentioned such as much lower, much shorter, and sufficiently close should be further elaborated. The details will be discussed elsewhere.

Taking into account that parameterized approximation spaces are examples of parameterized granules, one can generalize the above example of parameterized approximation space granulation to the case of granulation of parameterized granules.

In the process of searching for (sub-)optimal approximation spaces, different strategies are used. Let us consider an example of such strategies [309]. In the example, $DT = (U,A,d)$ denotes a decision system (a given sample of data), where

U is a set of objects, A is a set of attributes and d is a decision. We assume that for any object x, there is accessible only partial information equal to the A-signature of x (object signature, for short), *that is*, $Inf_A(x) = \{(a,a(x)); a \in A\}$ and analogously for any concept there is only given a partial information about this concept by a sample of objects, *for example*, in the form of decision system. One can use object signatures as new objects in a new relational structure $\mathcal{R}$. In this relational structure, $\mathcal{R}$ are also modelled some relations between object signatures, *for example*, defined by the similarities of these object signatures. Discovery of relevant relations on object signatures is an important step in the searching process for relevant approximation spaces. In this way, a class of relational structures representing perception of objects and their parts is constructed. In the next step, we select a language $\mathcal{L}$ of formulas expressing properties over the defined relational structures, and we search for relevant formulas in $\mathcal{L}$. The semantics of formulas (*e.g.* with one free variable) from $\mathcal{L}$ are subsets of object signatures. Observe that each object signature defines a neighbourhood of objects from a given sample (*e.g.* decision system DT) and another set on the whole universe of objects being an extension of U. In this way, each formula from $\mathcal{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. Such families can be used to define new neighbourhoods of a new approximation space, *for example*, by taking unions of the described above families. In the searching process for relevant neighbourhoods, we use information encoded in the given sample. More relevant neighbourhoods make it possible to define relevant approximation spaces (from the point of view of the optimization criterion). It is worth to mention that often this searching process is even more compound. For example, one can discover several relational structures (not only one, *e.g.* $\mathcal{R}$ as it was presented before) and formulas over such structures defining different families of neighbourhoods from the original approximation space and next fuse them for obtaining one family of neighbourhoods or one neighbourhood in a new approximation space. This kind of modelling is typical for hierarchical modelling [13], *for example*, when we search for a relevant approximation space for objects composed from parts for which some relevant approximation spaces have been already found.

3.5.3 Rough Sets and Higher Order Vagueness

In [111], it is stressed that vague concepts should have non-crisp boundaries. In the definition presented in this chapter, the notion of boundary region is defined as a crisp set $\mathsf{BN}_B(X)$. However, let us observe that this definition is relative to the subjective knowledge expressed by attributes from B. Different sources of information may use different sets of attributes for concept approximation. Hence, the boundary region can change when we consider these different views. Another aspect is discussed in [287, 307] where it is assumed that information about concepts is incomplete, *for example*, the concepts are given only on samples (see, *e.g.* [97, 115, 151]). From [287, 307], it follows that vague concepts cannot be approximated with

satisfactory quality by *static* constructs such as induced membership inclusion functions, approximations or models derived, *for example*, from a sample. Understanding of vague concepts can be only realized in a process in which the induced models are adaptively matching the concepts in a dynamically changing environment. This conclusion seems to have important consequences for further development of rough set theory in combination with fuzzy sets and other soft computing paradigms for adaptive approximate reasoning.

3.6 Information Granulation

Information granulation can be viewed as a human way of achieving data compression and it plays a key role in the implementation of the strategy of divide-and-conquer in human problem-solving [386, 223]. Objects obtained as the result of granulation are information granules. Examples of elementary information granules are indiscernibility or tolerance (similarity) classes (see, *e.g.* [222]). In reasoning about data and knowledge under uncertainty and imprecision many other more compound information granules are used (see, *e.g.* [255, 254, 284, 298, 299]). Examples of such granules are decision rules, sets of decision rules or classifiers. More compound information granules are defined by means of less compound ones. Note that inclusion or closeness measures between information granules should be considered rather than their strict equality. Such measures are also defined recursively for information granules.

Let us discuss shortly an example of information granulation in the process of modelling patterns for compound concept approximation (see, *e.g.* [21, 22, 24, 25, 28, 177, 310, 303, 313, 312], [332]. We start from a generalization of information systems. For any attribute $a \in A$ of an information system (U,A), we consider together with the value set V_a of a a relational structure $\mathcal{R}_a$ over the universe V_a (see, *e.g.* [309]). We also consider a language $\mathcal{L}_a$ of formulas (of the same relational signature as $\mathcal{R}_a$). Such formulas interpreted over $\mathcal{R}_a$ define subsets of Cartesian products of V_a. For example, any formula α with one free variable defines a subset $\|\alpha\|_{\mathcal{R}_a}$ of V_a. Let us observe that the relational structure $\mathcal{R}_a$ (without functions) induces a relational structure over U. Indeed, for any k-ary relation r from $\mathcal{R}_a$ one can define a k-ary relation $g_a \subseteq U^k$ by $(x_1,\ldots,x_k) \in g_a$ if and only if $(a(x_1),\ldots,a(x_k)) \in r$ for any $(x_1,\ldots,x_k) \in U^k$. Hence, one can consider any formula from $\mathcal{L}_a$ as a constructive method of defining a subset of the universe U with a structure induced by $\mathcal{R}_a$. Any such a structure is a new information granule. On the next level of hierarchical modelling, that is, in constructing new information systems we use such structures as objects and attributes are properties of such structures. Next, one can consider similarity between new constructed objects and then their similarity neighbourhoods will correspond to clusters of relational structures. This process is usually more complex. This is because instead of relational structure $\mathcal{R}_a$ we usually consider a fusion of relational structures corresponding to some attributes from A. The fusion makes it possible to describe constraints that should hold between parts obtained by

composition from less compound parts. Examples of relational structures can be defined by indiscernibility, similarity, intervals obtained in discretization or symbolic value grouping, preference or spatio-temporal relations (see, *e.g.* [76, 115, 297]). One can see that parameters to be tuned in searching for relevant[12] patterns over new information systems are, among others, relational structures over value sets, the language of formulas defining parts, and constraints.

3.7 Ontological Framework for Approximation

In a number of papers (see, *e.g.* [13, 193, 300, 14]), the problem of ontology approximation has been discussed together with possible applications to approximation of compound concepts or to knowledge transfer (see, *e.g.* [13, 17, 180, 276, 286, 300, 14]). For software RoughICE supporting ontology approximation the reader is referred to the system homepage[13].

In the ontology [328], (vague) concepts and local dependencies between them are specified. Global dependencies can be derived from local dependencies. Such derivations can be used as hints in searching for relevant compound patterns (information granules) in approximation of more compound concepts from the ontology. The ontology approximation problem is one of the fundamental problems related to approximate reasoning in distributed environments. One should construct (in a given language that is different from the ontology specification language) not only approximations of concepts from ontology but also vague dependencies specified in the ontology. It is worthwhile to mention that an ontology approximation should be induced on the basis of incomplete information about concepts and dependencies specified in the ontology. Information granule calculi based on rough sets have been proposed as tools making it possible to solve this problem. Vague dependencies have vague concepts in premisses and conclusions. The approach to approximation of vague dependencies based only on degrees of closeness of concepts from dependencies and their approximations (classifiers) is not satisfactory for approximate reasoning. Hence, more advanced approach should be developed. Approximation of any vague dependency is a method which allows for any object to compute the arguments "for" and "against" its membership to the dependency conclusion on the basis of the analogous arguments relative to the dependency premisses. Any argument is a compound information granule (compound pattern). Arguments are fused by local schemes (production rules) discovered from data. Further fusions are possible through composition of local schemes called approximate reasoning schemes (AR schemes) (see, *e.g.* [25, 199, 254]). To estimate the degree to which (at least) an object belongs to concepts from ontology, the arguments "for" and "against" those concepts are collected and next a conflict resolution strategy is applied to them to predict the degree.

[12] For target concept approximation.

[13] `http://www.mimuw.edu.pl/~bazan/roughice/`

3.8 Discernibility and Boolean Reasoning: Rough Set Methods for Machine Learning, Pattern Recognition, and Data Mining

Tasks collected under the labels of data mining, knowledge discovery, decision support, pattern classification, and approximate reasoning require tools aimed at discovering *templates (patterns)* in data and classifying them into certain *decision classes*. Templates are in many cases most frequent sequences of events, most probable events, regular configurations of objects, the decision rules of high quality, standard reasoning schemes. Tools for discovery and classification of templates are based on *reasoning schemes* rooted in various paradigms [51]. Such patterns can be extracted from data by means of methods based, *for example*, on Boolean reasoning and discernibility (see this Section and [36]).

Discernibility relations belong to the most important relations considered in rough set theory. The ability to discern between perceived objects is important for constructing many entities like reducts, decision rules or decision algorithms. In the classical rough set approach, a discernibility relation $DIS(B) \subseteq U \times U$ where $B \subseteq A$ is a subset of attributes of an information system (U,A), is defined by $xDIS(B)y$ if and only if $non(xIND_By)$, where IND_B is the B-indiscernibility relation. However, this is, in general, not the case for the generalized approximation spaces. One can define indiscernibility by $x \in \mathcal{I}(y)$ and discernibility by $\mathcal{I}(x) \cap \mathcal{I}(y) = \varnothing$ for any objects x,y, where $\mathcal{I}(x) = B(x), \mathcal{I}(y) = B(y)$ in the case of the indiscernibility relation, and $\mathcal{I}(x), \mathcal{I}(y)$ are neighborhoods of objects not necessarily defined by the equivalence relation in a more general case.

The idea of Boolean reasoning is based on construction for a given problem P of a corresponding Boolean function f_P with the following property: The solutions for the problem P can be decoded from prime implicants of the Boolean function f_P. Let us mention that to solve real-life problems it is necessary to deal with Boolean functions having large number of variables.

A successful methodology based on discernibility of objects and Boolean reasoning has been developed for computing of many entities important for applications, like reducts and their approximations, decision rules, association rules, discretization of real value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher-order surfaces, pattern extraction from data as well as conflict resolution or negotiation.

Most of the problems related to generation of the mentioned above entities are NP-complete or NP-hard. However, it was possible to develop efficient heuristics returning suboptimal solutions of the problems. The results of experiments on many data sets are very promising. They show very good quality of solutions generated by the heuristics in comparison with other methods reported in the literature (*e.g.* with respect to the classification quality of unseen objects). Moreover, these heuristics are very efficient from the point of view of time necessary for computing of solutions. Many of these methods are based on discernibility matrices. Note that it

is possible to compute the necessary information about these matrices using[14] information encoded in decision systems (*e.g.* sorted in preprocessing [16, 178, 373]) directly, which significantly improves the efficiency of algorithms.

It is important to note that the methodology makes it possible to construct heuristics having a very important *approximation property* which can be formulated as follows: Expressions generated by heuristics, that is, implicants *close* to prime implicants define approximate solutions for the problem.

In this section, we discuss applications of methods based on rough sets and Boolean reasoning in machine learning, pattern recognition and data mining.

In supervised machine learning paradigm [97, 115, 143, 151], a learning algorithm is given a training data set, usually in the form of a decision system $\mathbb{A} = (U, A, d)$[15], prepared by an expert. Every such decision system classifies elements from U into decision classes. The purpose of the algorithm is to return a set of decision rules together with matching procedure and conflict resolution strategy, called a classifier, which makes it possible to classify unseen objects, that is, objects that are not described in the original decision table. In this section, we provide a number of rough set methods that can be used in construction of classifiers. For more information, the reader is referred, *for example*, to [3, 20, 41, 42, 47, 50, 55, 64, 65, 71, 73, 85, 86, 87, 88, 89, 90, 92, 93, 98, 100, 101, 102, 113, 114, 125, 126, 130, 131, 138, 139, 140, 152, 153, 163, 176, 8, 194, 195, 196, 197, 199, 200, 218, 242, 249, 252, 253, 257, 270, 274, 275, 281, 283, 291, 293, 330, 342, 343, 344, 345, 347, 357, 369, 371, 374]), and for papers on hierarchical learning and ontology approximation, *for example*, to [17, 21, 24, 25, 177, 180, 179, 285, 294, 299, 300].

Most of the techniques discussed below are based on computing prime implicants for computing different kinds of reducts. Unfortunately, they are computationally hard. However, many heuristics have been developed which turned out to be very promising. The results of experiments on many data sets, reported in the literature, show a very good quality of classification of unseen objects using these heuristics. A variety of methods for computing reducts and their applications can be found in [16, 121, 135, 200, 252, 253, 281, 283, 293, 295, 315, 317, 374, 375]. The fact that the problem of finding a minimal reduct of a given information system is NP-hard was proved in [295].

As we mentioned, there exists a number of good heuristics that compute sufficiently many reducts in an acceptable time. Moreover, a successful methodology, based on different reducts, has been developed for solution of many problems like attribute selection, decision rule generation, association rule generation, discretization of real-valued attributes and symbolic value grouping. For further readings the reader is referred to [20, 280, 345] (attribute selection); [173, 164, 165, 171, 289] (discretization); [166, 167] (discretization of data stored in relational databases); and [172] (reduct approximation and association rules).

[14] That is, without the necessity of generation and storing of the discernibility matrices.

[15] For simplicity, we consider decision systems with one decision.

Many of these methods are based on discernibility matrices defined in this section. It is possible to compute the necessary information about these matrices using information or decision systems (*e.g.* sorted in preprocessing [16, 178]) directly what significantly improves the efficiency of algorithms.

The results presented in this section have been implemented, *for example*, in the RSES[16] software system (see also [15, 16, 19, 27, 116]). Sections 3.8.1-3.8.6 are based on a chapter of the book [48]. For links to other rough set software systems, the reader is referred to the RSDS[17].

3.8.1 Reducts in Information and Decision Systems

A crucial concept in the rough set approach to machine learning is that of a reduct. In fact, the term "reduct" corresponds to a wide class of concepts. What typifies all of them is that they are used to reduce information (decision) systems by removing redundant attributes. In this section, we consider three kinds of reducts that will be used in the remainder of this chapter.

Given an information system $\mathbb{A} = (U, A)$, a *reduct* is a minimal set (wrt inclusion) of attributes $B \subseteq A$ such that $IND_B = IND_A$ where IND_B, IND_A are the indiscernibility relations defined by B and A, respectively [215]. The intersection of all reducts is called a *core*.

Intuitively, a reduct is a minimal set of attributes from A that preserves the original classification defined by A. Reducts are extremely valuable in applications. Unfortunately, the problem of finding a minimal reduct is NP-hard. One can also show that for any m, there is an information system with m attributes having an exponential (wrt m) number of reducts. Fortunately, there are reasonably good heuristics that allow one to compute sufficiently many reducts in an acceptable amount of time.

To provide a general method for computing reducts, we will use the following constructs.

Let $\mathbb{A} = (U, A)$ be an information system with n objects. The *discernibility matrix* of $\mathbb{A}$ is an $n \times n$ matrix with elements c_{ij} consisting of the set of attributes from A on which objects x_i and x_j differ, that is,

$$c_{ij} = \{a \in A \ : \ a(x_i) \neq a(x_j)\}, \text{ for } i, j = 1, ..., n. \tag{3.37}$$

A *discernibility function* $f_{\mathbb{A}}$ for $\mathbb{A}$ is a propositional formula of m Boolean variables, $a_1^*, ..., a_m^*$, corresponding to the attributes $a_1, ..., a_m$, defined by

$$f_{\mathbb{A}}(a_1^*, ..., a_m^*) = \bigwedge_{1 \leq j < i \leq m} \ \bigvee_{c \in c_{ij}^*, c_{ij} \neq \varnothing} c, \tag{3.38}$$

where $c_{ij}^* = \{a^* \ : \ a \in c_{ij}\}$. In the sequel, we write a_i instead of a_i^*, for simplicity.

[16] The Rough Set Explorer System: `http://logic.mimuw.edu.pl/~rses/`

[17] The Rough Set Database System: `http://rsds.univ.rzeszow.pl`

Table 3.1. The information table considered in Example 3.3

Object	*Speed*	*Colour*	*Humidity*
car1	medium	green	high
car2	medium	yellow	low
car3	high	blue	high

Table 3.2. The discernibility matrix for the information table provided in Table 3.1

$\mathcal{M}(\mathbb{A})$	car1	car2	car3
car1		c,h	s,c
car2	c,h		s,c,h
car3	s,c	s,c,h	

The discernibility function $f_{\mathbb{A}}$ describes constraints that must hold to preserve discernibility between all pairs of discernible objects from $\mathbb{A}$. It requires keeping at least one attribute from each non-empty element of the discernibility matrix corresponding to any pair of discernible objects.

It can be shown [295] that for any information system $\mathbb{A} = (U,A)$ the set of all prime implicants of $f_{\mathbb{A}}$ determines the set of all reducts of $\mathbb{A}$.

Example 3.3. Consider the information system $\mathbb{A}$ whose associated information table is provided in Table 3.1. The discernibility matrix for $\mathbb{A}$ is presented in Table 3.2. (The letters s, c and h stand for $Speed$, $Color$ and $Humidity$, respectively.) The discernibility function for the information system $\mathbb{A}$ is then given by

$$f_{\mathbb{A}}(s,c,h) \equiv (c \vee h) \wedge (s \vee c) \wedge (s \vee c \vee h).$$

The prime implicants of $f_{\mathbb{A}}(s,c,h)$ can be computed in order to derive the reducts for $\mathbb{A}$:

$$\begin{aligned} f_{\mathbb{A}}(s,c,h) &\equiv (c \vee h) \wedge (s \vee c) \wedge (s \vee c \vee h) \\ &\equiv (c \vee h) \wedge (s \vee c) \\ &\equiv c \vee (h \wedge s). \end{aligned}$$

The prime implicants of $f_{\mathbb{A}}(s,c,h)$ are c and $h \wedge s$. Accordingly, there are two reducts of $\mathbb{A}$, namely $\{Color\}$ and $\{Humidity, Speed\}$. □

The second type of reduct used in this chapter are the *decision-relative reducts* for decision systems.

In terms of decision tables, $\partial_A(x)$, called the generalized decision function, is the mapping on U such that for any object x it specifies all rows in the table whose attribute values are the same as for x, and then collects the decision values from each row. A *decision-relative reduct* of $\mathbb{A} = (U,A,d)$ is a minimal (wrt inclusion) non-empty set of attributes $B \subseteq A$ such that $\partial_B = \partial_A$. Intuitively, the definition states that B allows us to classify exactly the same objects, as belonging to equivalence

Table 3.3. The decision table considered in Example 3.4

Object	*Speed*	*Colour*	*Humidity*	*Danger*
car1	medium	green	high	no
car2	medium	yellow	small	no
car3	high	blue	high	yes

classes U/∂_A, as A. In terms of decision tables, the columns associated with the attributes $A-B$ may be removed without affecting the classification power of the original table.

To compute decision-relative reducts, we extend the definitions of discernibility matrix and discernibility function in the following straightforward manner. Let $\mathbb{A} = (U,A,d)$ be a consistent decision system (i.e. $\partial_A(x)$ consists of exactly one decision for any $x \in U$) and let $\mathcal{M}(\mathbb{A}) = [c_{ij}]$ be the discernibility matrix of the information system (U,A). We construct a new matrix, $\mathcal{M}'(\mathbb{A}) = [c'_{ij}]$, where

$$c'_{ij} = \begin{cases} \varnothing, & \text{if } d(x_i) = d(x_j), \\ c_{ij}, & \text{otherwise.} \end{cases}$$

$\mathcal{M}'(\mathbb{A})$ is called the *decision-relative discernibility matrix* of $\mathbb{A}$. The *decision-relative discernibility function* $f^r_{\mathbb{A}}$ for $\mathbb{A}$ is constructed from the decision-relative discernibility matrix for $\mathbb{A}$ in the same way as a discernibility function is constructed from a discernibility matrix. Then, it can be shown [295] that the set of all prime implicants of $f^r_{\mathbb{A}}$ determines the set of all decision-relative reducts of the consistent decision system $\mathbb{A}$.

Example 3.4. Consider the decision table associated with a decision system $\mathbb{A}$ as represented in Table 3.3.

The discernibility matrix for $\mathbb{A}$ is the same as the one given in Table 3.2, and the decision-relative discernibility matrix for $\mathbb{A}$ is provided in Table 3.4.

Using the decision-relative discernibility matrix, we can compute the decision-relative discernibility function for $\mathbb{A}$:

$$f^r_{\mathbb{A}}(s,c,h) \equiv (s \vee c) \wedge (s \vee c \vee h) \equiv (s \vee c).$$

The set of all prime implicants of $f^r_{\mathbb{A}}(s,c,h)$ is $\{s,c\}$. Therefore, there are two decision-relative reducts of $\mathbb{A}$, namely $\{Speed\}$ and $\{Color\}$.

To each decision-relative reduct B of a decision system $\mathbb{A}$, we assign a new decision system, called the *B-reduction* of $\mathbb{A}$. The details are as follows. Let $\mathbb{A} = (U,A,d)$ be a consistent decision system and suppose that B is a decision-relative reduct of $\mathbb{A}$. A *B-reduction* of $\mathbb{A}$ is a decision system $\mathbb{A}^* = (V,B,d)$, where:[18]

- $V = \{[x]_B : x \in U\}$;

[18] Recall that $[x]_B$, where $x \in U$, denotes the equivalence class of the relation IND_B which contains x.

Table 3.4. The decision-relative discernibility matrix corresponding to the decision system shown in Table 3.3

$\mathcal{M}'(\mathbb{A})$	car1	car2	car3
car1			s,c
car2			s,c,h
car3	s,c	s,c,h	

Table 3.5. $\{Speed\}$-reduction of the decision system $\mathbb{A}$

Objects	*Speed*	*Danger*
car1, car2	medium	no
car3	high	yes

- $a([x]_B) = a(x)$, for each $a \in B$ and each $[x]_B \in V$;
- $d([x]_B) = d(x)$, for each $[x]_B \in V$.

Let $\mathbb{A}^*$ be the $\{Speed\}$-reduction of the decision system $\mathbb{A}$. The decision table associated with $\mathbb{A}^*$ is provided in Table 3.5. □

The above defined method for decision relative reducts computation can be easily extended to inconsistent decision systems.

Observe that another kind of reducts can be obtained by using the discernibility requirement relative to the positive regions, that is, $POS_A(d) = POS_B(d)$ instead of $\partial_B = \partial_A$. Certainly, for inconsistent decision systems the former requirement is less restrictive than the latter.

The last type of reduct, considered in this section, is used in applications where approximations to reducts are prefered to standard reducts. For example, approximate reducts for decision-relative reducts are making it possible to generate approximate decision rules. In the case of approximate reducts we relax the requirement for the discernibility preserving. Instead of preserving the discernibility for all entries of the discernibility matrix where it is necessary we preserve it to a degree, that is, in a number of entries characterized by a coefficient α. Such reducts are called α-reducts, where α is a real number from the interval $[0,1]$. More formal definition of approximate reducts is the following:

Let $\mathbb{A} = (U,A,d)$ be a decision system and let $\mathcal{M}(\mathbb{A})$ be the discernibility matrix of $\mathbb{A}$. Assume further that n is the number of non-empty sets in $\mathcal{M}(\mathbb{A})$. A set of attributes $B \subseteq A$ is called an α-*reduct* if and only if $\frac{m}{n} \geq \alpha$, where m is the number of sets in $\mathcal{M}(\mathbb{A})$ having a non-empty intersection with B.

The reader is referred to [169, 175, 215, 315, 317] for information on various types of approximate reducts. Additionally, [18, 172, 257, 318] provide approximation criteria based on discernibility and, therefore, related to Boolean reasoning principles.

3.8.2 Attribute Selection

In the supervised machine learning approach, a learning algorithm is provided with training data. In the context of rough set machine learning techniques, training data sets are provided in the form of training decision systems, or their equivalent representations as decision tables.

Since the condition attributes of a specific decision table are typically extracted from large sets of unstructured data, it is often the case that some of the attributes are irrelevant for the purpose of classification. Such attributes should be removed from the table if possible. The *attribute selection problem* is the problem of choosing a relevant subset of attributes, while removing the irrelevant ones (see, for example, [97]).

A natural solution of the attribute selection problem is to assume that the intersection of the decision-relative reducts of a training decision table is the source of the relevant attributes. Unfortunately, there are two problems with this solution. Firstly, the intersection can be empty. Secondly, the number of attributes contained in all decision-relative reducts is usually small. Consequently, although these attributes perfectly characterize the training decision table, they are, in general, inadequate for providing a satisfactory classification of new objects not occurring in the training data.

To deal with the attribute selection problem, it is often reasonable to use various approximations of decision-relative reducts.

Let $\mathbb{A} = (U, A, d)$ be a consistent decision system. One can treat any subset B of A as an *approximate reduct* of $\mathbb{A}$. The number

$$\varepsilon_{A,\{d\}}(B) = \frac{\gamma(A,\{d\}) - \gamma(B,\{d\})}{\gamma(A,\{d\})} = 1 - \frac{\gamma(B,\{d\})}{\gamma(A,\{d\})}, \tag{3.39}$$

is called an *error approximation* of A by B.[19]

The error approximation of A by B expresses exactly how the set of attributes B approximates the set of condition attributes A with respect to determination of d. Note that $\varepsilon_{A,\{d\}}(B) \in [0,1]$, where 0 indicates no error, and the closer $\varepsilon_{A,\{d\}}(B)$ is to 1, the greater is the error. The reader is referred, *for example*, to [172, 318] for more information on approximate reducts.

There are two general approaches to attribute selection: an open-loop approach and a closed-loop approach. Methods based on the open-loop approach are characterized by the fact that they do not use any feedback information about classifier quality for attribute selection. In contrast, the methods based on the closed-loop approach use feedback information as criteria for attribute selection.

A number of attribute selection algorithms have been proposed in the machine learning literature, but they will not be considered here since our focus is on rough set based techniques. Rough set techniques that attempt to solve the attribute

[19] Recall that the coefficient $\gamma(X,Y)$ expresses the degree of dependency between sets of attributes X and Y.

selection problem are typically based on the closed-loop approach and consist of the following basic steps:[20]

1. Decision-relative reducts are extracted from a training decision table. The attributes contained in these reducts (in their intersection or in most of them) are viewed as potentially relevant.
2. Using the specific machine learning algorithm, a classifier based on the chosen attributes is constructed.
3. The classifier is then tested on a new set of training data; if its performance is unsatisfactory (wrt some measure), a new set of attributes is constructed by extracting additional (approximate) reducts for the initial training table, and the process is repeated.

Reducts need not be the only source of information used in the selection of attributes. The rough set approach offers another interesting possibility. The main idea is to generalize the notion of attribute reduction by introducing the concept of *significance of attributes*. This measure enables attributes to be evaluated using a multi-valued scale that assigns a real number from the interval [0,1] to an attribute. This number, expressing the importance of an attribute in a decision system, is evaluated by measuring the effect of removing the attribute from the table.

The *significance of an attribute a* in a decision table $\mathbb{A} = (U, A, d)$ is defined by

$$\sigma_{A,\{d\}}(a) = \frac{\gamma(A,\{d\}) - \gamma(A-\{a\},\{d\})}{\gamma(A,\{d\})} = 1 - \frac{\gamma(A-\{a\},\{d\})}{\gamma(A,\{d\})}. \tag{3.40}$$

Assume that $B \subseteq A$. The significance coefficient can be extended to sets of attributes as follows,

$$\sigma_{(A,\{d\})}(B) = \frac{\gamma(A,\{d\}) - \gamma(A-B,\{d\})}{\gamma(A,\{d\})} = 1 - \frac{\gamma(A-B,\{d\})}{\gamma(A,\{d\})}. \tag{3.41}$$

The coefficient $\sigma_{A,\{d\}}(B)$ can be understood as a classification error which occurs when the attributes from B are removed from the decision system. Note that $\sigma_{A,\{d\}}(B) \in [0,1]$, where 0 indicates that removal of attributes in B causes no error, and the closer $\sigma_{A,\{d\}}(B)$ is to 1, the greater the error is.

Remark 3.1. In this section, we have mainly concentrated on the case, where the attributes are selected from the set of attributes of the input decision system. In some cases it might be useful to replace some attributes by a new one.

For example, if one considers a concept of a safe distance between vehicles, then attributes, say VS standing for "vehicle speed" and SL standing for "speed limit", can be replaced by an attribute DIF representing the difference $SL - VS$. In fact, the new attribute better corresponds to the concept of safe distance than the pair (VS, SL).

[20] There are public domain software packages, for instance the RSES system (for references see, *e.g.* [27] and http://logic.mimuw.edu.pl/~rses/), which offer software that may be used to solve the attribute selection problem.

3.8.3 Value Set Reduction

Consider a decision system with a large number of attribute values. There is a very low probability that a new object will be properly recognized by matching its attribute value vector with any of the rows in the decision table associated with the decision system. So, in order to construct a high-quality classifier, it is often necessary to reduce the cardinality of the value sets of specific attributes in a training decision table. The task of reducing the cardinality of value sets is referred to as the *value set reduction problem*(see, e.g. [169]).

In this section, two methods of value set reduction are considered [169]:

1. discretization, used for real value attributes, and
2. symbolic attribute value grouping, used for symbolic attributes.

3.8.3.1 Discretization

A discretization replaces value sets of condition real-valued attributes with intervals. The replacement ensures that a consistent decision system is obtained (assuming a given consistent decision system) by substituting original values of objects in the decision table by the unique names of the intervals comprising these values. This substantially reduces the size of the value sets of real-valued attributes.

The use of discretization is not specific to the rough set approach but to machine learning. In fact, a majority of rule or tree induction algorithms require it for a good performance.

Let $\mathbb{A} = (U, A, d)$ be a consistent decision system. Assume $V_a = [l_a, r_a) \subseteq \mathbb{R}$,[21] for any $a \in A$, and $l_a < r_a$. A pair (a, c), where $a \in A$ and $c \in V_a$, is called a *cut* on V_a.

Any attribute $a \in A$ defines a sequence of real numbers $v_1^a < v_2^a < \cdots < v_{k_a}^a$, where $\{v_1^a, v_2^a, \ldots, v_{k_a}^a\} = \{a(x) : x \in U\}$. The *set of basic cuts* on a, written B_a, is specified by

$$B_a = \{(a, (v_1^a + v_2^a)/2), (a, (v_2^a + v_3^a)/2), \ldots, (a, (v_{k_a-1}^a + v_{k_a}^a)/2)\}.$$

The set $\bigcup_{a \in A} B_a$ is called the *set of basic cuts* on $\mathbb{A}$. □

Example 3.5. Consider a consistent decision system $\mathbb{A}$ and the associated decision table presented in Table 3.6(a).

We assume that the initial value domains for the attributes a and b are

$$V_a = [0, 2); V_b = [0, 4).$$

The sets of values of a and b for objects from U are

[21] $\mathbb{R}$ denotes the set of real numbers.

Table 3.6. The discretization process: (a) The original decision system $\mathbb{A}$ considered in Example 3.5 (b) The C-discretization of $\mathbb{A}$ considered in Example 3.6

$\mathbb{A}$	a	b	d
u_1	0.8	2.0	1
u_2	1.0	0.5	0
u_3	1.3	3.0	0
u_4	1.4	1.0	1
u_5	1.4	2.0	0
u_6	1.6	3.0	1
u_7	1.3	1.0	1

(a)

$\Rightarrow$

$\mathbb{A}^C$	a^C	b^C	d
u_1	0	2	1
u_2	1	0	0
u_3	2	3	0
u_4	3	1	1
u_5	3	2	0
u_6	4	3	1
u_7	2	1	1

(b)

$$a(U) = \{0.8, 1.0, 1.3, 1.4, 1.6\};$$
$$b(U) = \{0.5, 1.0, 2.0, 3.0\}.$$

By definition, the sets of basic cuts for a and b are

$$B_a = \{(a,0.9),\ (a,1.15),\ (a,1.35),\ (a,1.5)\};$$
$$B_b = \{(b,0.75);\ (b,1.5);\ (b,2.5)\}.$$

□

Using the idea of cuts, decision systems with real-valued attributes can be discretized. For a decision system $\mathbb{A} = (U,A,d)$ and $a \in A$, let

$$C_a = \{(a,c_1^a),(a,c_2^a),\ldots,(a,c_k^a)\},$$

be any set of cuts of a. Assume that $c_1^a < c_2^a < \cdots < c_k^a$. The set of cuts $C = \bigcup_{a\in A} C_a$ defines a new decision system $\mathbb{A}^{\mathrm{C}} = (U,A^{\mathrm{C}},d)$, called the *C-discretization of* $\mathbb{A}$, where

- $A^{\mathrm{C}} = \{a^{\mathrm{C}} : a \in A\}$;
- $a^C(x) = \begin{cases} 0, & \text{if and only if } a(x) < c_1^a, \\ i, & \text{if and only if } a(x) \in [c_i^a, c_{i+1}^a), \text{ for } i \in \{1,\ldots,k-1\}, \\ k+1, & \text{if and only if } a(x) > c_k^a. \end{cases}$

Example 3.6 (Example 3.5 continued). Let $C = B_a \cup B_b$. It is easy to check that the C-discretization of $\mathbb{A}$ is the decision system whose decision table is provided in Table 3.6 (b). □

Since a decision system can be discretized in many ways, a natural question arises how to evaluate various possible discretizations.

A set of cuts C is called $\mathbb{A}$-*consistent*, if $\partial_{\mathbb{A}} = \partial_{\mathbb{A}^{\mathrm{C}}}$, where $\partial_{\mathbb{A}}$ and $\partial_{\mathbb{A}^{\mathrm{C}}}$ are generalized decision functions for $\mathbb{A}$ and $\mathbb{A}^{\mathrm{C}}$, respectively. An $\mathbb{A}$-consistent set of cuts C is $\mathbb{A}$-*irreducible* if C' is not $\mathbb{A}$-consistent for any $C' \subset C$. The $\mathbb{A}$-consistent set of cuts C is $\mathbb{A}$-*optimal* if $card(C) \leq card(C')$, for any $\mathbb{A}$-consistent set of cuts C'.

As easily observed, the set of cuts considered in Example 3.6 is $\mathbb{A}$-consistent. However, as we shall see in Example 3.7, it is neither optimal nor irreducible.

Since the purpose of the discretization process is to reduce the size of individual value sets of attributes, we are primarily interested in optimal sets of cuts. These are extracted from the basic sets of cuts for a given decision system.

Let $\mathbb{A} = (U,A,d)$ be a consistent decision system where $U = \{u_1,\ldots,u_n\}$. Recall that any attribute $a \in A$ defines a sequence $v_1^a < v_2^a < \cdots < v_{k_a}^a$, where $\{v_1^a, v_2^a, \ldots, v_{k_a}^a\} = \{a(x) : x \in U\}$. Let $ID(\mathbb{A})$ be the set of pairs (i,j) such that $i < j$ and $d(u_i) \neq d(u_j)$. We now construct a propositional formula, called the *discernibility formula* of $\mathbb{A}$, as follows:

1. To each interval of the form $\left[v_k^a, v_{k+1}^a\right)$, $a \in A$ and $k \in \{1,\ldots,n_a - 1\}$, we assign a Boolean variable denoted by p_k^a. The set of all these variables is denoted by $V(\mathbb{A})$.
2. We first construct a family of formulas

$$\{B(a,i,j) : a \in A \text{ and } (i,j) \in ID(\mathbb{A})\},$$

   where $B(a,i,j)$ is a disjunction of all elements from the set

$$\left\{p_k^a : \left[v_k^a, v_{k+1}^a\right) \subseteq [\min\{a(u_i),a(u_j)\}, \max\{a(u_i),a(u_j)\})\right\}.$$

3. Next, we construct a family of formulas

$$\{C(i,j) : i,j \in \{1,\ldots,n\}, i < j \text{ and } (i,j) \in ID(\mathbb{A})\},$$

   where $C(i,j) = \bigvee_{a \in A} B(a,i,j)$.
4. Finally, the discernibility formula for $\mathbb{A}$, $D(\mathbb{A})$, is defined as

$$D(\mathbb{A}) = \bigwedge C(i,j),$$

   where $i < j$ and $(i,j) \in ID(\mathbb{A})$ and $C(i,j) \not\equiv \text{FALSE}$.

Any non empty set $S = \{p_{k_1}^{a_1}, \ldots, p_{k_r}^{a_r}\}$ of Boolean variables from $V(\mathbb{A})$ uniquely defines a set of cuts, $C(S)$, given by

$$C(S) = \{(a_1, (v_{k_1}^{a_1} + v_{k_1+1}^{a_1})/2), \cdots, (a_r, (v_{k_r}^{a_r} + v_{k_r+1}^{a_r})/2)\}.$$

Then we have the following properties:

Let $\mathbb{A} = (U,A,d)$ be a consistent decision system. For any non-empty set $S \subseteq V(\mathbb{A})$ of Boolean variables, the following two conditions are equivalent:

1. The conjunction of variables from S is a prime implicant of the discernibility formula for $\mathbb{A}$.
2. $C(S)$ is an $\mathbb{A}$-irreducible set of cuts on $\mathbb{A}$. □

Let $\mathbb{A} = (U,A,d)$ be a consistent decision system. For any non-empty set $S \subseteq V(\mathbb{A})$ of Boolean variables, the following two conditions are equivalent:

1. The conjunction of variables from S is a minimal (wrt to length) prime implicant of the discernibility formula for $\mathbb{A}$.
2. $C(S)$ is an $\mathbb{A}$-optimal set of cuts on $\mathbb{A}$.

Example 3.7 (Example 3.6 continued).

$$ID(\mathbb{A}) = \{(1,2),\ (1,3),\ (1,5),\ (2,4),\ (2,6),\ (2,7)\\ (3,4),\ (3,6),\ (3,7),\ (4,5),\ (5,6),\ (5,7)\}.$$

1. We introduce four Boolean variables, p_1^a, p_2^a, p_3^a, p_4^a, corresponding respectively to the intervals

$$[0.8, 1.0),\ [1.0, 1.3),\ [1.3, 1.4),\ [1.4, 1.6)$$

of the attribute a, and three Boolean variables, p_1^b, p_2^b, p_3^b, corresponding respectively to the intervals

$$[0.5, 1.0),\ [1.0, 2.0),\ [2, 3.0)$$

of the attribute b
2. The following are the formulas $B(a,i,j)$ and $B(b,i,j)$, where $i < j$ and $(i,j) \in ID(\mathbb{A})$:

$$\begin{array}{ll}
B(a,1,2) \equiv p_1^a & B(b,1,2) \equiv p_1^b \vee p_2^b \\
B(a,1,3) \equiv p_1^a \vee p_2^a & B(b,1,3) \equiv p_3^b \\
B(a,1,5) \equiv p_1^a \vee p_2^a \vee p_3^a & B(b,1,5) \equiv \text{FALSE} \\
B(a,2,4) \equiv p_2^a \vee p_3^a & B(b,2,4) \equiv p_1^b \\
B(a,2,6) \equiv p_2^a \vee p_3^a \vee p_4^a & B(b,2,6) \equiv p_1^b \vee p_2^b \vee p_3^b \\
B(a,2,7) \equiv p_2^a & B(b,2,7) \equiv p_1^b \\
B(a,3,4) \equiv p_3^a & B(b,3,4) \equiv p_2^b \vee p_3^b \\
B(a,3,6) \equiv p_3^a \vee p_4^a & B(b,3,6) \equiv \text{FALSE} \\
B(a,3,7) \equiv \text{FALSE} & B(b,3,7) \equiv p_2^b \vee p_3^b \\
B(a,4,5) \equiv \text{FALSE} & B(b,4,5) \equiv p_2^b \\
B(a,5,6) \equiv p_4^a & B(b,5,6) \equiv p_3^b \\
B(a,5,7) \equiv p_3^a & B(b,5,7) \equiv p_2^b.
\end{array}$$

3. The following are the formulas $C(i,j)$, where $i < j$ and $(i,j) \in ID(\mathbb{A})$:

$$\begin{array}{ll}
C(1,2) \equiv p_1^a \vee p_1^b \vee p_2^b & C(1,3) \equiv p_1^a \vee p_2^a \vee p_3^b \\
C(1,5) \equiv p_1^a \vee p_2^a \vee p_3^a & C(2,4) \equiv p_2^a \vee p_3^a \vee p_1^b \\
C(2,6) \equiv p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b & C(2,7) \equiv p_2^a \vee p_1^b \\
C(3,4) \equiv p_3^a \vee p_2^b \vee p_3^b & C(3,6) \equiv p_3^a \vee p_4^a \\
C(3,7) \equiv p_2^b \vee p_3^b & C(4,5) \equiv p_2^b \\
C(5,6) \equiv p_4^a \vee p_3^b & C(5,7) \equiv p_3^a \vee p_2^b.
\end{array}$$

Table 3.7. The C-discretization considered in Example 3.7

$\mathbb{A}^C$	a^C	b^C	d
u_1	0	2	1
u_2	1	0	0
u_3	1	2	0
u_4	1	1	1
u_5	1	2	0
u_6	2	2	1
u_7	1	1	1

4. The discernibility formula for $\mathbb{A}$ is then given by

$$\begin{aligned} D(\mathbb{A}) \equiv\ & (p_1^a \vee p_1^b \vee p_2^b) \wedge (p_1^a \vee p_2^a \vee p_3^b) \wedge \\ & (p_1^a \vee p_2^a \vee p_3^a) \wedge (p_2^a \vee p_3^a \vee p_1^b) \wedge \\ & (p_2^a \vee p_3^a \vee p_4^a \vee p_1^b \vee p_2^b \vee p_3^b) \wedge (p_2^a \vee p_1^b) \wedge \\ & (p_3^a \vee p_2^b \vee p_3^b) \wedge (p_3^a \vee p_4^a) \wedge (p_2^b \vee p_3^b) \wedge \\ & p_2^b \wedge (p_4^a \vee p_3^b) \wedge (p_3^a \vee p_2^b). \end{aligned}$$

The prime implicants of the formula $D(\mathbb{A})$ are

$$\begin{aligned} & p_2^a \wedge p_4^a \wedge p_2^b \\ & p_2^a \wedge p_3^a \wedge p_2^b \wedge p_3^b \\ & p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b \\ & p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b. \end{aligned}$$

Suppose we take the prime implicant $p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b$. Its corresponding set of cuts is

$$C = \{(a, 0.9), (a, 1.5), (b, 0.75), (b, 1.5)\}.$$

The decision table for the C-discretization of $\mathbb{A}$ is provided in Table 3.7.

Observe that the set of cuts corresponding to the prime implicant $p_2^a \wedge p_4^a \wedge p_2^b$ is $\{(a, 1.15), (a, 1.5), (b, 1.5)\}$. Thus C is not an optimal set of cuts. □

The problem of searching for an optimal set of cuts P in a given decision system $\mathbb{A}$ is NP-hard. However, it is possible to devise efficient heuristics which, in general, return reasonable sets of cuts. One of them, called MD-heuristics, is presented below.

We say that a cut (a, c) discerns objects x and y if and only if $a(x) < c \leq a(y)$ or $a(y) < c \leq a(x)$.

Let n be the number of objects and let k be the number of attributes of a decision system $\mathbb{A}$. It can be shown that the best cut can be found in $O(kn)$ steps using $O(kn)$ space only.

Example 3.8. Consider the decision table with the associated decision system $\mathbb{A}$, provided in Table 3.6 from Example 3.5. The associated information table for the information system $\mathbb{A}^*$ is presented in Table 3.8.

INPUT: a decision system $\mathbb{A} = (U, A, d)$
OUTPUT: a set of cuts $\mathcal{C}$

1. Set $\mathcal{C}$ to $\varnothing$.
2. Let $\bigcup_{a \in A} C_a$ be the set of basic cuts on $\mathbb{A}$.
3. Construct an information table $\mathbb{A}^* = (U^*, A^*)$ such that
   - U^* is the set of pairs (u_i, u_j) of objects discerned by d (in $\mathbb{A}$) such that $i < j$;
   - $A^* = \bigcup_{a \in A} C_a$, where for each $c \in A^*$,

$$c(x,y) = \begin{cases} 1, & \text{if and only if } c \text{ discerns } x \text{ and } y \text{ (in } \mathbb{A}), \\ 0, & \text{otherwise.} \end{cases}$$

4. Choose a column from $\mathbb{A}^*$ with the maximal number of occurrences of 1's; add the cut corresponding to this column to $\mathcal{C}$; delete the column from $\mathbb{A}^*$, together with all rows marked with 1 in it.
5. If A^* is non-empty, then go to step 4 else stop. □

Table 3.8. The information table for the information system $\mathbb{A}^*$

$\mathbb{A}^*$	$(a, 0.9)$	$(a, 1.15)$	$(a, 1.35)$	$(a, 1.5)$	$(b, 0.75)$	$(b, 1.5)$	$(b, 2.5)$
(u_1, u_2)	1	0	0	0	1	1	0
(u_1, u_3)	1	1	0	0	0	0	1
(u_1, u_5)	1	1	1	0	0	0	0
(u_2, u_4)	0	1	1	0	1	0	0
(u_2, u_6)	0	1	1	1	1	1	1
(u_2, u_7)	0	1	0	0	1	0	0
(u_3, u_4)	0	0	1	0	0	1	1
(u_3, u_6)	0	0	1	1	0	0	0
(u_3, u_7)	0	0	0	0	0	1	1
(u_4, u_5)	0	0	0	0	0	1	0
(u_5, u_6)	0	0	0	1	0	0	1
(u_5, u_7)	0	0	1	0	0	1	0

Under the assumption that columns with maximal number of 1's are chosen from left to right (if many such columns exist in a given step), the set of cuts returned by the algorithm is $\{(a, 1.35), (b, 1.5), (a, 1.15), (a, 1.5)\}$. However, as shown in example 3.7, it is not an optimal set of cuts. □

3.8.3.2 Symbolic Attribute Value Grouping

Symbolic attribute value grouping is a technique for reducing the cardinality of value sets of symbolic attributes. Let $\mathbb{A} = (U, A, d)$ be a decision system. Any function $c_a : V_a \to \{1, \ldots, m\}$, where $m \leq card(V_a)$, is called a *clustering function* for V_a. The *rank of* c_a, denoted by $rank(c_a)$, is the value $card(\{c_a(x) \mid x \in V_a\})$.

For $B \subseteq A$, a family of clustering functions $\{c_a\}_{a\in B}$ is *B-consistent* if and only if

$$\forall a \in B \ [c_a(a(u)) = c_a(a(u'))],$$

implies $$(u,u') \in \mathcal{IND}_B \cup \mathcal{IND}_d, \text{ for any pair } (u,u') \in U.$$

The notion of B-consistency has the following intuitive interpretation: If two objects are indiscernible wrt clustering functions for value sets of attributes from B, then they are indiscernible either by the attributes from B or by the decision attribute.

We consider the following problem, called the *symbolic value partition grouping problem*:

Given a decision system $\mathbb{A} = (U,A,d)$, where $U = \{u_1,\ldots,u_k\}$, and a set of attributes $B \subseteq A$, search for a B-consistent family $\{c_a\}_{a\in B}$ of clustering functions such that $\sum_{a\in B} rank\,(c_a)$ is minimal.

In order to solve this problem, we apply the following steps:

1. Introduce a set of new Boolean variables:[22]

   $$\{a_v^{v'} :\ a \in B \text{ and } v,v' \in V_a \text{ and } v \neq v'\}.$$

   We extract a subset S of this set such that $a_v^{v'} \in S$ implies that $v' < v$ wrt some arbitrary linear order $<$ on the considered domain.
2. Construct matrix $\mathcal{M} = [c_{ij}]_{i,j=1,\ldots,k}$ as follows:

   $$c_{ij} = \{a_v^{v'} \in S :\ v' = a(u_i) \text{ and } v = a(u_j) \text{ and } d(u_i) \neq d(u_j)\}.$$

   It is easily seen that in the case of a binary decision, the matrix can be reduced by placing objects corresponding to the first decision in rows and those corresponding to the second decision in columns. We call such a matrix a *reduced discernibility matrix.*
3. Using the reduced matrix, $\mathcal{M}'$, obtained in the previous step, construct the function

   $$\bigwedge_{c_{ij}\in\mathcal{M}'} \left(\bigvee_{c\in c_{ij}, c_{ij}\neq\varnothing} c \right).$$

4. Compute the shortest (or sufficiently short) prime implicant I of the constructed function.
5. Using I, construct, for each attribute $a \in B$, an undirected graph $\Gamma_a = \langle V_a^\Gamma, E_a^\Gamma \rangle$, where

   - $V_a^\Gamma = \{a_v \mid v \in V_a\}$;
   - $E_a^\Gamma = \{(a_x,a_y) \mid x,y \in U \text{ and } a(x) \neq a(y)\}$.
     Note that using I one can construct E_a^Γ due to the equality

     $$E_a^\Gamma = \{(a_v,a_{v'}) : a_v^{v'} \text{ occurs in } I\}.$$

[22] The introduced variables serve to discern between pairs of objects wrt an attribute a.

Table 3.9. The decision table considered in Example 3.9

$\mathcal{A}$	a	b	d
u_1	a_1	b_1	0
u_2	a_1	b_2	0
u_3	a_2	b_3	0
u_4	a_3	b_1	0
u_5	a_1	b_4	1
u_6	a_2	b_2	1
u_7	a_2	b_1	1
u_8	a_4	b_2	1
u_9	a_3	b_4	1
u_{10}	a_2	b_5	1

6. Find a minimal colouring of vertices for Γ_a.[23] The colouring defines a partition of V_a^Γ by assuming that all vertices of the same colour belong to the same partition set and no partition set contains vertices with different colours. Partition sets are named using successive natural numbers.
   The clustering function for V_a^Γ is $c_a(a_v) = i$, provided that a_v is a member of the i-th partition set.

Remark 3.2. In practical implementations, one does not usually construct the matrix $\mathcal{M}$ explicitly, as required in Steps (2)-(3) above. Instead, prime implicants are directly extracted from the original decision system.

It should be emphasized that in Step (4) above, there can be many different shortest prime implicants and in Step (6) there can be many different colorings of the obtained graphs. Accordingly, one can obtain many substantially different families of clustering functions resulting in different classifiers. In practice, one often generates a number of families of clustering functions, tests them against data and chooses the best one.

Using the construction above to generate a family of partitions, it is usually possible to obtain a substantially smaller decision table, according to the following definition.

Let $\mathbb{A} = (U, A, d)$ be a decision system and $B \subseteq A$. Any family of clustering functions $c = \{c_a\}_{a \in B}$ specifies a new decision system $\mathbb{A}^c = (U, A^c, d)$ called the *c-reduction of* $\mathbb{A}$ wrt B, where $A^c = \{a^{\mathbf{c}} : a \in B\}$ and $a^c(x) = c_a(a(x))$.

Example 3.9. Consider the decision table provided in Table 3.9. The goal is to solve the symbolic value partition problem for $B = A$.

One then has to perform the following steps:

1. Introduce new Boolean variables a_v^u, b_x^w, for all $u, v \in V_a, u < v$ and $w, x \in V_b, w < x$.

[23] The colourability problem is solvable in polynomial time for $k = 2$, but remains NP-complete for all $k \geq 3$. But, similarly to discretization, one can apply some efficient search heuristics for generating (sub-) optimal partitions.

Table 3.10. The reduced matrix corresponding to the decision table provided in Table 3.9

$\mathcal{M}'$	u_1	u_2	u_3	u_4
u_5	$b^{b_1}_{b_4}$	$b^{b_2}_{b_4}$	$a^{a_1}_{a_2}, b^{b_3}_{b_4}$	$a^{a_1}_{a_3}, b^{b_1}_{b_4}$
u_6	$a^{a_1}_{a_2}, b^{b_1}_{b_2}$	$a^{a_1}_{a_2}$	$b^{b_2}_{b_3}$	$a^{a_2}_{a_3}, b^{b_1}_{b_2}$
u_7	$a^{a_1}_{a_2}$	$a^{a_1}_{a_2}, b^{b_1}_{b_2}$	$b^{b_1}_{b_3}$	$a^{a_2}_{a_3}$
u_8	$a^{a_1}_{a_4}, b^{b_1}_{b_2}$	$a^{a_1}_{a_4}$	$a^{a_2}_{a_4}, b^{b_2}_{b_3}$	$a^{a_3}_{a_4}, b^{b_1}_{b_2}$
u_9	$a^{a_1}_{a_3}, b^{b_1}_{b_4}$	$a^{a_1}_{a_3}, b^{b_2}_{b_4}$	$a^{a_2}_{a_3}, b^{b_3}_{b_4}$	$b^{b_1}_{b_4}$
u_{10}	$a^{a_1}_{a_2}, b^{b_1}_{b_5}$	$a^{a_1}_{a_2}, b^{b_2}_{b_5}$	$b^{b_3}_{b_5}$	$a^{a_2}_{a_3}, b^{b_1}_{b_5}$

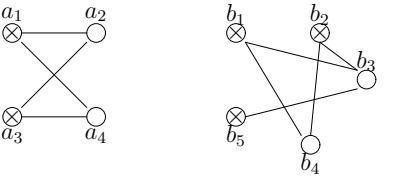


Fig. 3.3. Coloring of attribute value graphs constructed in Example 3.9

Table 3.11. The reduced table corresponding to graphs shown in Figure 3.3

a^c	b^c	d
1	1	0
2	2	0
1	2	1
2	1	1

2. The reduced matrix $\mathcal{M}'$ is presented in Table 3.10.
3. The required Boolean function is given by

$$\begin{aligned} & b^{b_1}_{b_4} \wedge b^{b_2}_{b_4} \wedge (a^{a_1}_{a_2} \vee b^{b_3}_{b_4}) \wedge (a^{a_1}_{a_3} \vee b^{b_1}_{b_4}) \wedge \\ & (a^{a_1}_{a_2} \vee b^{b_1}_{b_2}) \wedge a^{a_1}_{a_2} \wedge b^{b_2}_{b_3} \wedge (a^{a_2}_{a_3} \vee b^{b_1}_{b_2}) \wedge \\ & a^{a_1}_{a_2} \wedge (a^{a_1}_{a_2} \vee b^{b_1}_{b_2}) \wedge b^{b_1}_{b_3} \wedge a^{a_2}_{a_3} \wedge \\ & (a^{a_1}_{a_4} \vee b^{b_1}_{b_2}) \wedge a^{a_1}_{a_4} \wedge (a^{a_2}_{a_4} \vee b^{b_2}_{b_3}) \wedge (a^{a_3}_{a_4} \vee b^{b_1}_{b_2}) \wedge \\ & (a^{a_1}_{a_3} \vee b^{b_1}_{b_4}) \wedge (a^{a_1}_{a_3} \vee b^{b_2}_{b_4}) \wedge (a^{a_2}_{a_3} \vee b^{b_3}_{b_4}) \wedge b^{b_1}_{b_4} \wedge \\ & (a^{a_1}_{a_2} \vee b^{b_1}_{b_5}) \wedge (a^{a_1}_{a_2} \vee b^{b_2}_{b_5}) \wedge b^{b_3}_{b_5} \wedge (a^{a_2}_{a_3} \vee b^{b_1}_{b_5}). \end{aligned}$$

4. The shortest prime implicant for the function is

$$I \equiv a^{a_1}_{a_2} \wedge a^{a_2}_{a_3} \wedge a^{a_1}_{a_4} \wedge a^{a_3}_{a_4} \wedge b^{b_1}_{b_4} \wedge b^{b_2}_{b_4} \wedge b^{b_2}_{b_3} \wedge b^{b_1}_{b_3} \wedge b^{b_3}_{b_5}.$$

5. The graphs corresponding to a and b are shown in Figure 3.3.
6. The graphs are 2-colored, as shown in Figure 3.3, where nodes marked by $\otimes$ are colored black and the other nodes are colored white. These colorings generate the following clustering functions:

$$\begin{aligned} c_a(a_1) &= c_a(a_3) = 1 \\ c_a(a_2) &= c_a(a_4) = 2 \\ c_b(b_1) &= c_b(b_2) = c_b(b_5) = 1 \\ c_b(b_3) &= c_b(b_4) = 2. \end{aligned}$$

Given these clustering functions, one can construct a new decision system (see Table 3.11). □

Observe that discretization and symbolic attribute value grouping can be simultaneously used in decision systems including both real-value and symbolic attributes.

3.8.4 Minimal Decision Rules

In this section, techniques for constructing minimal rules for decision systems will be considered.

Given a decision Table $\mathbb{A}$, a *minimal decision rule* (wrt $\mathbb{A}$) is a rule which is TRUE in $\mathbb{A}$ and which becomes FALSE in $\mathbb{A}$ if any elementary descriptor from the left-hand side of the rule is removed.[24]

The minimal number of elementary descriptors in the left-hand side of a minimal decision rule defines the largest subset of a decision class. Accordingly, information included in the conditional part of any minimal decision rule is sufficient for predicting the decision value of all objects satisfying this part of the rule. The conditional parts of minimal decision rules define the largest object sets relevant for approximating decision classes. The conditional parts of minimal decision rules can be computed using prime implicants.

To compute the set of all minimal rules wrt to a decision system $\mathbb{A} = (U, A, d)$, we proceed as follows, for any object $x \in U$:

1. Construct a decision-relative discernibility function f_x^r by considering the row corresponding to object x in the decision-relative discernibility matrix for $\mathbb{A}$.
2. Compute all prime implicants of f_x^r.
3. On the basis of the prime implicants, create minimal rules corresponding to x. To do this, consider the set $A(I)$ of attributes corresponding to propositional variables in I, for each prime implicant I, and construct the rule:

$$\left(\bigwedge_{a \in A(I)} (a = a(x)) \right) \Rightarrow d = d(x).$$

[24] A decision rule $\varphi \Rightarrow \psi$ is TRUE in $\mathbb{A}$ if and only if $\|\varphi\|_{\mathbb{A}} \subseteq \|\psi\|_{\mathbb{A}}$.

Table 3.12. Decision table considered in Example 3.10

Object	L	W	C	S
1	7.0	large	green	no
2	7.0	large	blue	no
3	4.0	medium	green	yes
4	4.0	medium	red	yes
5	5.0	medium	blue	no
6	4.5	medium	green	no
7	4.0	large	red	no

Table 3.13. $\{L,W\}$-reduction considered in Example 3.10

Objects	L	W	S
1, 2	7.0	large	no
3,4	4.0	medium	yes
5	5.0	medium	no
6	4.5	medium	no
7	4.0	large	no

The following example illustrates the idea.

Example 3.10. Consider the decision system $\mathbb{A}$ whose decision table is provided in Table 3.12. Table 3.12 contains the values of condition attributes of vehicles (L,W,C, standing for *Length, Width*, and *Color*, respectively), and a decision attribute S standing for *Small* which allows one to decide whether a given vehicle is small.

This system has exactly one decision-relative reduct consisting of attributes L and W. The $\{L,W\}$-reduction of $\mathbb{A}$ as shown in Table 3.13.

Table 3.13 results in the following set of non-minimal decision rules:

$$(L = 7.0) \wedge (W = \mathsf{large}) \Rightarrow (S = \mathsf{no})$$
$$(L = 4.0) \wedge (W = \mathsf{medium}) \Rightarrow (S = \mathsf{yes})$$
$$(L = 5.0) \wedge (W = \mathsf{medium}) \Rightarrow (S = \mathsf{no})$$
$$(L = 4.5) \wedge (W = \mathsf{medium}) \Rightarrow (S = \mathsf{no})$$
$$(L = 4.0) \wedge (W = \mathsf{large}) \Rightarrow (S = \mathsf{no}).$$

To obtain the minimal decision rules, we apply the construction provided above, for $x \in \{1,\ldots,7\}$.

1. The decision-relative discernibility functions $f_1^r,\ldots,f_7^r$ are constructed on the basis of the reduced discernibility matrix shown in Table 3.14:

Table 3.14. Reduced decision-relative discernibility matrix from Example 3.10

	3	4
1	L,W	L,W,C
2	L,W,C	L,W,C
5	L,C	L,C
6	L	L,C
7	W,C	W

$$
\begin{aligned}
f_1^r &\equiv (L \vee W) \wedge (L \vee W \vee C) \equiv (L \vee W) \\
f_2^r &\equiv (L \vee W \vee C) \wedge (L \vee W \vee C) \equiv (L \vee W \vee C) \\
f_3^r &\equiv (L \vee W) \wedge (L \vee W \vee C) \wedge (L \vee C) \wedge L \wedge (W \vee C) \\
&\equiv (L \wedge W) \vee (L \wedge C) \\
f_4^r &\equiv (L \vee W \vee C) \wedge (L \vee W \vee C) \wedge (L \vee C) \wedge (L \vee C) \wedge W \\
&\equiv (L \wedge W) \vee (C \wedge W) \\
f_5^r &\equiv (L \vee C) \wedge (L \vee C) \equiv (L \vee C) \\
f_6^r &\equiv L \wedge (L \vee C) \equiv L \\
f_7^r &\equiv (W \vee C) \wedge W \equiv W.
\end{aligned}
$$

2. The following prime implicants are obtained from formulas $f_1^r, \ldots, f_7^r$:

f_1^r: L, W
f_2^r: L, W, C
f_3^r: $L \wedge W, L \wedge C$
f_4^r: $L \wedge W, C \wedge W$
f_5^r: L, C
f_6^r: L
f_7^r: W.

3. Based on the prime implicants, minimal decision rules are created for objects $1, \ldots, 7$. For instance, from prime implicants L and W corresponding to f_1^r, the following minimal decision rules are generated based on object 1:

$$
\begin{aligned}
&(L = 7.0) \Rightarrow (S = \mathsf{no}) \\
&(W = \mathsf{large}) \Rightarrow (S = \mathsf{no}).
\end{aligned}
$$

On the basis of object 3 and prime implicants $L \wedge W$ and $L \wedge C$ for f_3^r we obtain the following rules:

$$
\begin{aligned}
&(L = 4.0) \wedge (W = \mathsf{medium}) \Rightarrow (S = \mathsf{yes}) \\
&(L = 4.0) \wedge (C = \mathsf{green}) \Rightarrow (S = \mathsf{yes}).
\end{aligned}
$$

Similarly, minimal decision rules can easily be obtained for all other formulas. □

In practice, the number of minimal decision rules can be large. One then tries to consider only subsets of these rules or to drop some conditions from minimal rules.

Remark 3.3. The main challenge in inducing rules from decision systems lies in determining which attributes should be included into the conditional parts of the rules. Using the strategy outlined above, the minimal rules are computed first. Their conditional parts describe the largest object sets with the same generalized decision value in a given decision system. Although such minimal decision rules can be computed, this approach can result in a set of rules of unsatisfactory classification quality. Such rules might appear too general or too specific for classifying new objects. This depends on the data analyzed. Techniques have been developed for the further tuning of minimal rules. □

3.8.5 Example: Learning of Concepts

Given that one has all the techniques described in the previous sections at one's disposal, an important task is to induce definitions of concepts from training data, where the representation of the definition is as efficient and of high quality as possible. These definitions may then be used as classifiers for the induced concepts.

Let us concentrate on the concept of *Distance* between cars on the road. The rough relation $Distance(x,y,z)$ denotes the approximate distance between vehicles x and y, where $z \in \{\mathsf{small}, \mathsf{medium}, \mathsf{large}, \mathsf{unknown}\}$. Below, we simplify the definition somewhat, and consider $Distance(x,z)$ which denotes that the distance between x and the vehicle directly preceding x is z.[25] Assume that sample training data have been gathered in a decision table which is provided in Table 3.15, where[26]

- *SL* stands for the "speed limit" on a considered road segment;
- *VS* stands for the "vehicle speed";
- *W* stands for "weather conditions";
- *AD* stands for "actual distance" between a given vehicle and its predecessor on the road.

For the sake of simplicity, we concentrate on generating rules to determine whether the distance between two objects is small.

On the basis of the training data, one can compute a discernibility matrix. Since we are interested in rules for the decision small only, it suffices to consider a simplified discernibility matrix with columns labelled by objects 1 and 3, as these are the only two objects, where the corresponding decision is small. The resulting discernibility matrix is shown in Table 3.16.

[25] In fact, here we consider a distance to be small if it causes a dangerous situation and to be large if the situation is safe.

[26] Of course, real-life sample data would consist of hundreds or thousands of examples.

Table 3.15. Training data considered in Section 3.8.5

Object	*SL*	*VS*	*W*	*AD*	*Distance*
1	70	60	rain	3.0	small
2	70	70	sun	5.0	medium
3	50	60	rain	5.0	small
4	50	60	sun	9.0	medium
5	30	15	rain	9.0	large
6	30	10	sun	5.0	large
7	70	60	rain	15.0	large
8	50	40	rain	15.0	large

Table 3.16. Discernibility matrix of Table 3.15 for decision small

Object	1	3
2	VS, W, AD	SL, VS, W
4	SL, W, AD	W, AD
5	SL, VS, AD	SL, VS, AD
6	SL, VS, W, AD	SL, VS, W
7	AD	SL, AD
8	SL, VS, AD	VS, AD

The discernibility matrix gives rise to the following discernibility functions:

$$\begin{aligned} f_1 &\equiv (VS \vee W \vee AD) \wedge (SL \vee W \vee AD) \wedge (SL \vee VS \vee AD) \\ &\quad \wedge (SL \vee VS \vee W \vee AD) \wedge AD \wedge (SL \vee VS \vee AD) \\ &\equiv AD \\ f_3 &\equiv (SL \vee VS \vee W) \wedge (W \vee AD) \wedge (SL \vee VS \vee AD) \\ &\quad \wedge (SL \vee VS \vee W) \wedge (SL \vee AD) \wedge (VS \vee AD) \\ &\equiv (W \wedge AD) \vee (SL \wedge AD) \vee (VS \wedge AD) \vee (SL \wedge VS \wedge W). \end{aligned}$$

Based on the discernibility functions, one can easily find prime implicants and obtain the following rules for the decision small:[27]

$$\begin{aligned} &(AD = 3.0) \Rightarrow (Distance = \text{small}) \\ &(W = \text{rain}) \wedge (AD = 5.0) \Rightarrow (Distance = \text{small}) \\ &(SL = 50) \wedge (AD = 5.0) \Rightarrow (Distance = \text{small}) \\ &(VS = 60) \wedge (AD = 5.0) \Rightarrow (Distance = \text{small}) \\ &(SL = 50) \wedge (VS = 60) \wedge (W = \text{rain}) \Rightarrow (Distance = \text{small}). \end{aligned} \tag{3.42}$$

There have been also developed methods for approximation of compound concepts based on rough sets, hierarchical learning and ontology approximation (see, *e.g.* [13, 17, 21, 24, 25, 177, 180, 179, 285, 294, 299, 300]).

[27] In practical applications one would have to discretize *AD* before extracting rules.

Table 3.17. Information table considered in Example 3.11

Customer	*Bread*	*Milk*	*Jam*	*Beer*
1	yes	yes	no	no
2	yes	yes	yes	yes
3	yes	yes	yes	no
4	no	yes	yes	no

3.8.6 Association Rules

In this section [172, 175], we show how rough set techniques can be used to extract association rules from information systems. Association rules playing an important role in the field of data mining provide associations among attributes[28]. A real number from the interval [0,1] is assigned to each rule and provides a measure of the confidence of the rule. The following example will help to illustrate this.

Example 3.11. Consider the information table provided in Table 3.17.

Each row in the table represents items bought by a customer. For instance, customer 1 bought bread and milk, whereas customer 4 bought milk and jam. An association rule that can be extracted from the above table is: *a customer who bought bread also bought milk*. This is represented by

$$(Bread = \text{yes}) \Rightarrow (Milk = \text{yes}).$$

Since all customers who bought bread actually bought milk too, the confidence of this rule is 1. Now consider the rule

$$(Bread = \text{yes}) \wedge (Milk = \text{yes}) \Rightarrow (Jam = \text{yes})$$

stating that a customer who bought bread and milk, bought jam as well. Since three customers bought both bread and milk and two of them bought jam, the confidence of this rule is 2/3. □

We now formalize this approach to confidence measures for association rules. Recall that by a template, we mean a conjunction of elementary descriptors, that is, expressions of the form $a = v$, where a is an attribute and $v \in V_a$. For an information system $\mathbb{A}$ and a template T, we denote by $support_{\mathbb{A}}(T)$ the number of objects satisfying T. Let $\mathbb{A}$ be an information system and $T = D_1 \wedge \ldots \wedge D_m$ be a template. By an *association rule generated from* T, we mean any expression of the form

$$\bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \in Q} D_j,$$

where $\{P, Q\}$ is a partition of $\{D_1, \ldots, D_m\}$. By a *confidence of an association rule* $\phi \equiv \bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \in Q} D_j$ we mean the coefficient

[28] Association between attributes are also studied using association reducts [315].

$$confidence_{\mathbb{A}}(\phi) = \frac{support_{\mathbb{A}}(D_1 \wedge \ldots \wedge D_m)}{support_{\mathbb{A}}(\bigwedge\limits_{D_i \in P} D_i)}.$$

There are two basic steps used in methods aimed at generating association rules. (Below s and c stand for support and confidence thresholds wrt a given information system $\mathbb{A}$, respectively.)

1. Generate as many templates $T = D_1 \wedge \ldots \wedge D_k$ as possible, such that $support_{\mathbb{A}}(T) \geq s$ and $support_{\mathbb{A}}(T \wedge D_i) < s$, for any descriptor D_i different from all descriptors $D_1, \ldots, D_k$.
2. Search for a partition $\{P, Q\}$ of T, for each T generated in the previous step, satisfying

   a. $support_{\mathbb{A}}(P) < \frac{support_{\mathbb{A}}(T)}{c}$

   b. P has the shortest length among templates satisfying (a).

   Every such partition leads to an association rule of the form $P \Rightarrow Q$ whose confidence is greater than c.

The second step, crucial to the process of extracting association rules, can be solved using rough set methods.

Let $T = D_1 \wedge D_2 \wedge \ldots \wedge D_m$ be a template such that $support_{\mathbb{A}}(T) \geq s$. For a given confidence threshold $c \in [0,1]$, the association rule $\phi \equiv P \Rightarrow Q$ is called *c-irreducible* if $confidence_{\mathbb{A}}(P \Rightarrow Q) \geq c$ and for any association rule $\phi' \equiv P' \Rightarrow Q'$ such that P' is a sub-formula of P, we have

$$confidence_{\mathbb{A}}(P' \Rightarrow Q') < c.$$

The problem of searching for c-irreducible association rules from a given template is equivalent to the problem of searching for α-reducts in a decision table, for some $\alpha \in [0,1]$ (see Section 3.8.1).

Let $\mathbb{A}$ be an information system and $T = D_1 \wedge D_2 \wedge \ldots \wedge D_m$ be a template. By a *characteristic table* for T wrt $\mathbb{A}$, we understand a decision system $\mathbb{A}|_T = (U, A|_T, d)$, where

1. $A|_T = \{a_{D_1}, a_{D_2}, \ldots, a_{D_m}\}$ is a set of attributes corresponding to the descriptors of T such that

$$a_{D_i}(u) = \begin{cases} 1, \text{ if the object } u \text{ satisfies } D_i, \\ 0, \text{ otherwise;} \end{cases}$$

2. the decision attribute d determines whether the object satisfies a template T, that is,

$$d(u) = \begin{cases} 1, \text{ if the object } u \text{ satisfies } T, \\ 0, \text{ otherwise.} \end{cases}$$

The following property provides the relationship between association rules and approximations of reducts.

For a given information system $\mathbb{A} = (U, A)$, a template $T = D_1 \wedge D_2 \wedge \ldots \wedge D_m$ and a set of descriptors $P \subseteq \{D_1, \ldots, D_m\}$, the association rule

$$\bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \in \{D_1, \ldots, D_m\} - P} D_j,$$

is

1. a 1-irreducible association rule from T if and only if $\bigcup_{D_i \in P} \{a_{D_i}\}$ is a decision-relative reduct of $\mathbb{A}|_T$;
2. a c-irreducible association rule from T if and only if $\bigcup_{D_i \in P} \{a_{D_i}\}$ is an α-reduct of $\mathbb{A}|_T$, where

$$\alpha = 1 - \left[\left(\frac{1}{c} - 1 \right) \Big/ \left(\frac{|U|}{support_{\mathbb{A}}(T)} - 1 \right) \right].$$

The problem of searching for the shortest association rules is NP-hard.

The following example illustrates the main ideas used in the searching method for association rules.

Example 3.12. Consider the information table $\mathcal{A}$ with 18 objects and 9 attributes presented in Table 3.18.

Table 3.18. Information table $\mathbb{A}$ considered in Example 3.12

$\mathcal{A}$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
u_1	0	1	1	1	80	2	2	2	3
u_2	0	1	2	1	81	0	aa	1	aa
u_3	0	2	2	1	82	0	aa	1	aa
u_4	0	1	2	1	80	0	aa	1	aa
u_5	1	1	2	2	81	1	aa	1	aa
u_6	0	2	1	2	81	1	aa	1	aa
u_7	1	2	1	2	83	1	aa	1	aa
u_8	0	2	2	1	81	0	aa	1	aa
u_9	0	1	2	1	82	0	aa	1	aa
u_{10}	0	3	2	1	84	0	aa	1	aa
u_{11}	0	1	3	1	80	0	aa	2	aa
u_{12}	0	2	2	2	82	0	aa	2	aa
u_{13}	0	2	2	1	81	0	aa	1	aa
u_{14}	0	3	2	2	81	2	aa	2	aa
u_{15}	0	4	2	1	82	0	aa	1	aa
u_{16}	0	3	2	1	83	0	aa	1	aa
u_{17}	0	1	2	1	84	0	aa	1	aa
u_{18}	1	2	2	1	82	0	aa	2	aa

Consider the template

$$T = (a_1 = 0) \wedge (a_3 = 2) \wedge (a_4 = 1) \wedge (a_6 = 0) \wedge (a_8 = 1). \tag{3.43}$$

It is easily seen that $support_{\mathcal{A}}(T) = 10$. The new constructed decision table $\mathcal{A}|_T$ is presented in Table 3.19.

Table 3.19. Decision table $\mathbb{A}|_T$ considered in Example 3.12

$\mathcal{A}\|_T$	a_{D_1}	a_{D_2}	a_{D_3}	a_{D_4}	a_{D_5}	d
	$(a_1 = 0)$	$(a_3 = 2)$	$(a_4 = 1)$	$(a_6 = 0)$	$(a_8 = 1)$	
u_1	1	0	1	0	0	0
u_2	1	1	1	1	1	1
u_3	1	1	1	1	1	1
u_4	1	1	1	1	1	1
u_5	0	1	0	0	1	0
u_6	1	0	0	0	1	0
u_7	0	0	0	0	1	0
u_8	1	1	1	1	1	1
u_9	1	1	1	1	1	1
u_{10}	1	1	1	1	1	1
u_{11}	1	0	1	1	0	0
u_{12}	1	0	0	1	0	0
u_{13}	1	1	1	1	1	1
u_{14}	1	1	0	0	0	0
u_{15}	1	1	1	1	1	1
u_{16}	1	1	1	1	1	1
u_{17}	1	1	1	1	1	1
u_{18}	0	1	1	1	0	0

The reduced discernibility matrix $\mathcal{A}|_T$ is provided in Table 3.20, where for simplicity, the second column represents, in fact, ten columns with identical contents, labelled by $u_2, u_3, u_4, u_8, u_9, u_{10}, u_{13}, u_{15}, u_{16}, u_{17}$, respectively.

Table 3.20. Reduced discernibility matrix for $\mathcal{A}|_T$ from Example 3.12

$\mathcal{M}(\mathcal{A}\|_T)$	u_2, u_3, u_4, u_8, u_9 $u_{10}, u_{13}, u_{15}, u_{16}, u_{17}$
u_1	$a_{D_2}, a_{D_4}, a_{D_5}$
u_5	$a_{D_1}, a_{D_3}, a_{D_4}$
u_6	$a_{D_2}, a_{D_3}, a_{D_4}$
u_7	$a_{D_1}, a_{D_2}, a_{D_3}, a_{D_4}$
u_{11}	$a_{D_1}, a_{D_3}, a_{D_5}$
u_{12}	$a_{D_2}, a_{D_3}, a_{D_5}$
u_{14}	$a_{D_3}, a_{D_4}, a_{D_5}$
u_{18}	a_{D_1}, a_{D_5}

Given the discernibility matrix, one can easily compute the discernibility function $\mathcal{A}|_T$ for $\mathcal{A}|_T$:

$$\begin{aligned} f_{\mathbb{A}_T}(a_{D_1}, a_{D_2}, a_{D_3}, a_{D_4}, a_{D_5}) \equiv\ & (a_{D_2} \vee a_{D_4} \vee a_{D_5}) \\ & \wedge (a_{D_1} \vee a_{D_3} \vee a_{D_4}) \\ & \wedge (a_{D_2} \vee a_{D_3} \vee a_{D_4}) \\ & \wedge (a_{D_1} \vee a_{D_2} \vee a_{D_3} \vee a_{D_4}) \\ & \wedge (a_{D_1} \vee a_{D_3} \vee a_{D_5}) \\ & \wedge (a_{D_2} \vee a_{D_3} \vee a_{D_5}) \\ & \wedge (a_{D_3} \vee a_{D_4} \vee a_{D_5}) \\ & \wedge (a_{D_1} \vee a_{D_5}), \end{aligned}$$

where D_i denotes the i-th conjunct of (3.43).

The discernibility function has the following prime implicants: $a_{D_3} \wedge a_{D_5}$, $a_{D_4} \wedge a_{D_5}$, $a_{D_1} \wedge a_{D_2} \wedge a_{D_3}$, $a_{D_1} \wedge a_{D_2} \wedge a_{D_4}$, $a_{D_1} \wedge a_{D_2} \wedge a_{D_5}$, $a_{D_1} \wedge a_{D_3} \wedge a_{D_4}$. This gives rise to the reducts: $\{a_{D_3}, a_{D_5}\}$, $\{a_{D_4}, a_{D_5}\}$, $\{a_{D_1}, a_{D_2}, a_{D_3}\}$, $\{a_{D_1}, a_{D_2}, a_{D_4}\}$, $\{a_{D_1}, a_{D_2}, a_{D_5}\}$, $\{a_{D_1}, a_{D_3}, a_{D_4}\}$. Thus, there are 6 association rules with confidence 1, that is, 1-irreducible:

$$\begin{aligned} D_3 \wedge D_5 &\Rightarrow D_1 \wedge D_2 \wedge D_4 \\ D_4 \wedge D_5 &\Rightarrow D_1 \wedge D_2 \wedge D_3 \\ D_1 \wedge D_2 \wedge D_3 &\Rightarrow D_4 \wedge D_5 \\ D_1 \wedge D_2 \wedge D_4 &\Rightarrow D_3 \wedge D_5 \\ D_1 \wedge D_2 \wedge D_5 &\Rightarrow D_3 \wedge D_4 \\ D_1 \wedge D_3 \wedge D_4 &\Rightarrow D_2 \wedge D_5. \end{aligned}$$

For confidence 0.9, we look for α-reducts for the decision table $\mathcal{A}|_T$, where

$$\alpha = 1 - \left(\frac{1}{0.9} - 1\right) / \left(\frac{18}{10} - 1\right) \approx 0.86.$$

Hence, we look for a set of descriptors that covers at least $\lceil (18-10) * \alpha \rceil = \lceil 8 * 0.86 \rceil = 7$ elements of the discernibility matrix $\mathcal{M}(\mathcal{A}|_{\mathbf{T}})$. One can see that the following sets of descriptors: $\{D_1, D_2\}$, $\{D_1, D_3\}$, $\{D_1, D_4\}$, $\{D_1, D_5\}$, $\{D_2, D_3\}$, $\{D_2, D_5\}$, $\{D_3, D_4\}$ have non-empty intersections with exactly 7 members of the discernibility matrix $\mathcal{M}(\mathcal{A}|_T)$. Consequently, the 0.9-irreducible association rules obtained from those sets are the following:

$$
\begin{aligned}
D_1 \wedge D_2 &\Rightarrow D_3 \wedge D_4 \wedge D_5 \\
D_1 \wedge D_3 &\Rightarrow D_2 \wedge D_4 \wedge D_5 \\
D_1 \wedge D_4 &\Rightarrow D_2 \wedge D_3 \wedge D_5 \\
D_1 \wedge D_5 &\Rightarrow D_2 \wedge D_3 \wedge D_4 \\
D_2 \wedge D_3 &\Rightarrow D_1 \wedge D_4 \wedge D_5 \\
D_2 \wedge D_5 &\Rightarrow D_1 \wedge D_3 \wedge D_4 \\
D_3 \wedge D_4 &\Rightarrow D_1 \wedge D_2 \wedge D_5.
\end{aligned}
$$

The technique illustrated by this example can be applied to find useful dependencies between attributes in complex application domains. In particular, one could use such dependencies in constructing robust classifiers conforming to the laws of the underlying reality. □

3.9 Rough Sets, Approximate Boolean Reasoning and Scalability

Mining large data sets is one of the biggest challenges in KDD. In many practical applications, there is a need of data mining algorithms running on terminals of a client–server database system where the only access to database (located in the server) is enabled by SQL queries.

Unfortunately, the proposed so far data mining methods based on rough sets and Boolean reasoning approach are characterized by high computational complexity and their straightforward implementations are not applicable for large data sets. The critical factor for time complexity of algorithms solving the discussed problem is the number of simple SQL queries like

```
SELECT COUNT FROM aTable WHERE aCondition
```

In this section, we present some efficient modifications of these methods to solve out this problem. We consider the following issues:

- Searching for short reducts from large data sets;
- Searching for best partitions defined by cuts on continuous attributes;

3.9.1 Reduct Calculation

Let us again illustrate the idea of reduct calculation using discernibility matrix (Table 3.21).

Example 3.13. Let us consider the "weather" problem specified by decision system which is represented by decision table (see Table 3.21). Objects are described by four condition attributes and are divided into 2 classes. Let us consider the first 12

observations. In this example, $U = \{1,2,\ldots,12\}$, $A = \{a_1,a_2,a_3,a_4\}$, $CLASS_{no} = \{1,2,6,8\}$, $CLASS_{yes} = \{3,4,5,7,9,10,11,12\}$.

Table 3.21. The exemplary "weather" decision table (left) and the compact form of discernibility matrix (right)

date	outlook	temperature	humidity	windy	play
ID	a_1	a_2	a_3	a_4	*dec*
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes

$\mathcal{M}$	1	2	6	8
3	a_1	a_1,a_4	$a_1,a_2,$ a_3,a_4	a_1,a_2
4	a_1,a_2	a_1,a_2,a_4	a_2,a_3,a_4	a_1
5	a_1,a_2,a_3	$a_1,a_2,$ a_3,a_4	a_4	a_1,a_2,a_3
7	$a_1,a_2,$ a_3,a_4	a_1,a_2,a_3	a_1	$a_1,a_2,$ a_3,a_4
9	a_2,a_3	a_2,a_3,a_4	a_1,a_4	a_2,a_3
10	a_1,a_2,a_3	$a_1,a_2,$ a_3,a_4	a_2,a_4	a_1,a_3
11	a_2,a_3,a_4	a_2,a_3	a_1,a_2	a_3,a_4
12	a_1,a_2,a_4	a_1,a_2	a_1,a_2,a_3	a_1,a_4

The discernibility matrix can be treated as a board containing $n \times n$ boxes. Noteworthy is the fact that discernibility matrix is symmetrical with respect to the main diagonal, because $M_{i,j} = M_{j,i}$, and that sorting all objects according to their decision classes causes a shift off all empty boxes nearby to the main diagonal. In case of decision table with two decision classes, the discernibility matrix can be rewritten in a more compact form as shown in Table 3.21. The discernibility function is constructed from discernibility matrix by taking a conjunction of all discernibility clauses in which any attribute a_i is substituted by the corresponding Boolean variable x_i. After reducing, of all repeated clauses, we have[29]:

$$\begin{aligned} f(x_1,x_2,x_3,x_4) =& (x_1)(x_1+x_4)(x_1+x_2)(x_1+x_2+x_3+x_4)(x_1+x_2+x_4) \\ & (x_2+x_3+x_4)(x_1+x_2+x_3)(x_4)(x_2+x_3)(x_2+x_4) \\ & (x_1+x_3)(x_3+x_4)(x_1+x_2+x_4). \end{aligned}$$

One can find relative reducts of the decision table by searching for prime implicants of this discernibility function. The straightforward method allow us to calculate all prime implicants by transformation of the formula to the DNF form (using absorbtion rule $p(p+q) \equiv p$ and other rules for Boolean algebra). One can do it as follows:

$$f = (x_1)(x_4)(x_2+x_3) = x_1x_4x_2 + x_1x_4x_3$$

Thus, we have 2 reducts: $R_1 = \{a_1,a_2,a_4\}$ and $R_2 = \{a_1,a_3,a_4\}$.

[29] In the formulas $+$ denotes logical disjunction $\vee$ and we omit the conjunction sign $\wedge$ if is this not lead to misunderstanding.

Every heuristic algorithm for the prime implicant problem can be applied to the discernibility function to solve the minimal reduct problem. One of such heuristics was proposed in [295] and was based on the idea of greedy algorithm, where each attribute is evaluated by its discernibility measure, that is, the number of pairs of objects which are discerned by the attribute, or, equivalently, the number of its occurrences in the discernibility matrix.

- First we have to calculate the number of occurrences of each attributes in the discernibility matrix:

$$eval(a_1) = disc_{dec}(a_1) = 23, \qquad eval(a_2) = disc_{dec}(a_2) = 23,$$
$$eval(a_3) = disc_{dec}(a_3) = 18, \qquad eval(a_4) = disc_{dec}(a_4) = 16.$$

  Thus a_1 and a_2 are the two most preferred attributes.
- Assume that we select a_1. Now we are taking under consideration only those cells of the discernibility matrix that are not containing a_1. There are 9 such cells only, and the number of occurrences are as the following:

$$eval(a_2) = disc_{dec}(a_1,a_2) - disc_{dec}(a_1) = 7,$$
$$eval(a_3) = disc_{dec}(a_1,a_3) - disc_{dec}(a_1) = 7,$$
$$eval(a_4) = disc_{dec}(a_1,a_4) - disc_{dec}(a_1) = 6.$$

- If this time we select a_2, then there only 2 remaining cells, and, both are containing a_4;
- Therefore, the greedy algorithm returns the set $\{a_1,a_2,a_4\}$ as a reduct of sufficiently small size.

There is another reason for choosing a_1 and a_4, because they are *core attributes*[30]. One can check that an attribute is a core attribute if and only if occurs in the discernibility matrix as a singleton [295]. Therefore, core attributes can be recognized by searching for all singleton cells of the discernibility matrix. The pseudo-code of this algorithm is presented in Algorithm 3.1.

The reader may have a feeling that the greedy algorithm for reduct problem has quite a high complexity, because two main operations:

- $disc(B)$ – number of pairs of objects discerned by attributes from B;
- $isCore(a)$ – check whether a is a core attribute;

are defined by the discernibility matrix which is a complex data structure containing $O(n^2)$ cells, and each cell can contain up to $O(m)$ attributes, where n is the number of objects and m is the number of attributes of the given decision table. This suggests that the two main operations need at least $O(mn^2)$ computational time.

Fortunately, both operations can be performed more efficiently. It has been shown [178] that both operations can be calculated in time $O(mn\log n)$ without the necessity to store the discernibility matrix. We present an effective implementation of this heuristics that can be applied to large data sets.

[30] An attribute is called core attribute if and only if it occurs in every reduct [215, 222].

Algorithm 3.1. Searching for short reduct

begin
  $B := \varnothing$ // Step 1. Initializing B by core attributes
  for $a \in A$ **do**
    if $isCore(a)$ **then**
      $B := B \cup \{a\}$
    end
  end
  // Step 2. Including attributes to B
  repeat
    $a_{\max} := \arg\max_{a \in A-B} disc_{dec}(B \cup \{a\})$ $eval(a_{\max}) := disc_{dec}(B \cup \{a_{\max}\}) - disc_{dec}(B)$
    if $(eval(a_{\max}) > 0)$ **then**
      $B := B \cup \{a\}$
    end
  until $(eval(a_{\max}) == 0)$ *OR* $(B == A)$;
  // Step 3. Elimination
  for $a \in B$ **do**
    if $(disc_{dec}(B) = disc_{dec}(B - \{a\}))$ **then**
      $B := B - \{a\}$;
    end
  end
end

Let $\mathbb{A} = (U, A, dec)$ be a decision system. By a "*counting table*" of a set of objects $X \subset U$ we denote the vector:

$$CountTable(X) = \langle n_1, \ldots, n_d \rangle,$$

where $n_k = card(X \cap CLASS_k)$ is the number of objects from X belonging to the k^{th} decision class.

We define a conflict measure of X by

$$conflict(X) = \sum_{i<j} n_i n_j = \frac{1}{2}\left[\left(\sum_{k=1}^{d} n_k\right)^2 - \sum_{k=1}^{d} n_k^2\right].$$

In other words, $conflict(X)$ is the number of pairs $(x, y) \in X \times X$ of objects from different decision classes.

By a *counting table* of a set of attributes B we mean the two-dimensional array $Count(B) = [n_{v,k}]_{v \in INF(B), k \in V_{dec}}$, where

$$n_{v,k} = card(\{x \in U : inf_B(x) = v \text{ and } dec(x) = k\}).$$

Thus $Count(B)$ is a collection of counting tables of equivalence classes of the indiscernibility relation $IND(B)$. It is clear that the complexity time for the construction of counting table is $O(nd \log n)$, where n is the number of objects and d is the

number of decision classes. One can also observe that counting tables can be easily constructed in data base management systems using simple SQL queries.

For a given counting table, one can easily calculate the discernibility measure relative to a set of attributes B by

$$disc_{dec}(B) = \frac{1}{2} \sum_{v \neq v', k \neq k'} n_{v,k} \cdot n_{v',k'}.$$

The disadvantage of this equation relates to the fact that it requires $O(S^2)$ operations, where S is the size of the counting table $Count(B)$.

The discernibility measure can be understood as a number of unresolved (by the set of attributes B) conflicts. One can show that

$$disc_{dec}(B) = conflict(U) - \sum_{[x] \in U/IND(B)} conflict([x]_{IND(B)}). \tag{3.44}$$

Thus, the discernibility measure can be determined in $O(S)$ time:

$$disc_{dec}(B) = \frac{1}{2}\left(n^2 - \sum_{k=1}^{d} n_k^2\right) - \frac{1}{2} \sum_{v \in INF(B)} \left[\left(\sum_{k=1}^{d} n_{v,k}\right)^2 - \sum_{k=1}^{d} n_{v,k}^2\right], \tag{3.45}$$

where $n_k = |CLASS_k| = \sum_v n_{v,k}$ is the size of k^{th} decision class.

Moreover, one can show that attribute a is a core attribute of decision system $\mathbb{A} = (U, A, dec)$ if and only if

$$disc_{dec}(A - \{a\}) < disc_{dec}(A).$$

Thus both operations $disc_{dec}(B)$ and $isCore(a)$ can be performed in linear time with respect to the counting table size.

Example 3.14. In the discussed example, the counting table for a_1 is as follows:

$Count(a_1)$	$dec = no$	$dec = yes$
$a_1 = sunny$	3	2
$a_1 = overcast$	0	3
$a_1 = rainy$	1	3

We illustrate Eqn. (3.45) by inserting some additional columns to the counting table:

$Count(a_1)$	$dec = no$	$dec = yes$	$\sum$	$conflict(.)$
$a_1 = sunny$	3	2	5	$\frac{1}{2}(5^2 - 2^2 - 3^2) = 6$
$a_1 = overcast$	0	3	3	$\frac{1}{2}(3^2 - 0^2 - 3^2) = 0$
$a_1 = rainy$	1	3	4	$\frac{1}{2}(4^2 - 1^2 - 3^2) = 3$
U	4	8	12	$\frac{1}{2}(12^2 - 8^2 - 4^2) = 32$

Thus, $disc_{dec}(a_1) = 32 - 6 - 0 - 3 = 23$.

3.9.2 Discretization of Large Data Sets Stored in Relational Databases

In this section (see [169, 166, 167]), we discuss an application of approximate Boolean reasoning to efficient searching for cuts in large data sets stored in relational databases. Searching for relevant cuts is based on simple statistics which can be efficiently extracted from relational databases. This additional statistical knowledge is making it possible to perform the searching based on Boolean reasoning much more efficient. It can be shown that the extracted cuts by using such reasoning are quite close to optimal.

Searching algorithms for optimal partitions of real-valued attributes, defined by cuts, have been intensively studied. The main goal of such algorithms is to discover cuts that can be used to synthesize decision trees or decision rules of high quality wrt some quality measures (*e.g.* quality of classification of new unseen objects, quality defined by the decision tree height, support and confidence of decision rules).

In general, all those problems are hard from computational point of view (*e.g.* the searching problem for minimal and consistent set of cuts is NP-hard). In consequence, numerous heuristics have been developed for approximate solutions of these problems. These heuristics are based on approximate measures estimating the quality of extracted cuts. Among such measures *discernibility measures* are relevant for the rough set approach.

We outline an approach for solution of a searching problem for optimal partition of real-valued attributes by cuts, assuming that the large data table is represented in a relational database. In such a case, even the linear time complexity with respect to the number of cuts is not acceptable because of the time needed for one step. The critical factor for time complexity of algorithms solving that problem is the number of SQL queries of the form

```
SELECT COUNT
  FROM a Table
     WHERE (an attribute BETWEEN value1 AND value2)
           AND (additional condition)
```

necessary to construct partitions of real-valued attribute sets. We assume the answer time for such queries does not depend on the interval length[31]. Using a straightforward approach to optimal partition selection (wrt a given measure), the number of necessary queries is of order $O(N)$, where N is the number of preassumed cuts. By introducing some optimization measures, it is possible to reduce the size of searching space. Moreover, using only $O(\log N)$ simple queries, suffices to construct a partition very close to optimal.

Let $\mathbb{A} = (U, A, d)$ be a decision system with real-valued condition attributes. Any cut (a, c), where $a \in A$ and c is a real number, defines two disjoint sets given by

[31] This assumption is satisfied in some existing database management systems.

$$U_L(a,c) = \{x \in U : a(x) \leq c\},$$
$$U_R(a,c) = \{x \in U : a(x) > c\}.$$

If both $U_L(a,c)$ and $U_R(a,c)$ are non-empty, then c is called a a. The cut (a,c) discerns a pair of objects x, y if either $a(x) < c \leq a(y)$ or $a(y) < c \leq a(x)$.

Let $\mathbb{A} = (U,A,d)$ be a decision system with real-valued condition attributes and decision classes X_i, for $i = 1,\ldots,r(d)$. A *quality of a cut* (a,c), denoted by $W(a,c)$, is defined by

$$\begin{aligned} W(a,c) &= \sum_{i \neq j}^{r(d)} L_i(a,c) * R_j(a,c) \\ &= \left(\sum_{i=1}^{r(d)} L_i(a,c)\right) * \left(\sum_{i=1}^{r(d)} R_i(a,c)\right) - \sum_{i=1}^{r(d)} L_i(a,c) * R_i(a,c), \end{aligned} \tag{3.46}$$

where $L_i(a,c) = card(X_i \cap U_L(a,c))$ and $R_i(a,c) = card(X_i \cap U_R(a,c))$, for $i = 1,\ldots,r(d)$.

In the sequel, we will be interested in finding cuts maximizing the function $W(a,c)$.

The following definition will be useful. Let $\mathcal{C}_a = \{(a,c_1),\ldots,(a,c_N)\}$ be a set of cuts on attribute a, over a decision system $\mathbb{A}$ and assume $c_1 < c_2 ... < c_N$. By a *median of the* i^{th} *decision class*, denoted by $Median(i)$, we mean the minimal index j for which the cut $(a,c_j) \in \mathcal{C}_a$ minimizes the value $|L_i(a,c_j) - R_i(a,c_j)|$,[32] where L_i and R_i are defined before.

One can use only $O(r(d) * \log N)$ SQL queries to determine the medians of decision classes by using the well-known binary search algorithm.

Then one can show that the quality function $W_a(i) \stackrel{\text{def}}{=} W(a,c_i)$, for $i = 1,\ldots,N$, is increasing in $\{1,\ldots,min\}$ and decreasing in $\{max,\ldots,N\}$, where min and max are defined by

$$min = \min_{1 \leq i \leq N} Median(i),$$
$$max = \max_{1 \leq i \leq N} Median(i).$$

In consequence, the search space for maximum of $W(a,c_i)$ is reduced to $i \in [min,max]$.

Now, one can apply the divide and conquer strategy to determine the best cut, given by $c_{Best} \in [c_{min},c_{max}]$, wrt the chosen quality function. First, we divide the interval containing all possible cuts into k intervals. Using some heuristics, one then predict the interval which most probably contains the best cut. This process is recursively applied to that interval, until the considered interval consists of one cut. The problem which remains to be solved is how to define such approximate measures which could help us to predict the suitable interval.

[32] The minimization means that $|L_i(a,c_j) - R_i(a,c_j)| = \min_{1 \leq k \leq N} |L_i(a,c_k) - R_i(a,c_k)|$.

Let us consider a simple probabilistic model. Let (a,c_L), (a,c_R) be two cuts such that $c_L < c_R$ and $i = 1,\ldots,r(d)$. For any cut (a,c) satisfying $c_L < c < c_R$, we assume that $x_1,\ldots,x_{r(d)}$, where $x_i = card(X_i \cap U_L(a,c) \cap U_R(a,c))$ are independent random variables with uniform distribution over sets $\{0,...,M_1\}$, ..., $\{0,...,M_{r(d)}\}$, respectively, that

$$M_i = M_i(a,c_L,c_R) = card(X_i \cap U_L(a,c_R) \cap U_R(a,c_L)).$$

Under these assumptions, the following fact holds. For any cut $c \in [c_L,c_R]$, the mean $E(W(a,c))$ of quality $W(a,c)$ is given by

$$E(W(a,c)) = \frac{W(a,c_L) + W(a,c_R) + conflict((a,c_L),(a,c_R))}{2}, \tag{3.47}$$

where $conflict((a,c_L),(a,c_R)) = \sum_{i \neq j} M_i * M_j$.

In addition, the standard deviation of $W(a,c)$ is given by

$$D^2(W(a,c)) = \sum_{i=1}^{n} \left[\frac{M_i(M_i+2)}{12} \left(\sum_{j \neq i} (R_j(a,c_R) - L_j(a,c_L)) \right)^2 \right]. \tag{3.48}$$

Formulas (3.47) and (3.48) can be used to construct a predicting measure for the quality of the interval $[c_L,c_R]$:

$$Eval([c_L,c_R],\alpha) = E(W(a,c)) + \alpha\sqrt{D^2(W(a,c))}, \tag{3.49}$$

where the real parameter $\alpha \in [0,1]$ can be tuned in a learning process.

To determine the value $Eval([c_L,c_R],\alpha)$, we need to compute the numbers

$$L_1(a,c_L),\ldots,L_{r(d)}(a,c_L),M_1,\ldots,M_{r(d)},R_1(a,c_R),\ldots,R_{r(d)}(a,c_R).$$

This requires $O(r(d))$ SQL queries of the form

```
SELECT COUNT
FROM DecTable
WHERE (attribute a BETWEEN value1 AND value2)
       AND (dec = i).
```

Hence, the number of queries required for running this algorithm is

$$O(r(d)k\log_k N).$$

In practice, we set $k = 3$, since the function $f(k) = r(d)k\log_k N$ over positive integers is taking minimum for $k = 3$.

Numerous experiments on different data sets have shown that the proposed solution allows one to find a cut which is very close to the optimal one. For more details the reader is referred to the literature (see [166, 167]).

3.10 Rough Sets and Logic

The father of contemporary logic is a German mathematician Gottlob Frege (1848-1925). He thought that mathematics should not be based on the notion of set but on the notions of logic. He created the first axiomatized logical system but it was not understood by the logicians of those days.

During the first three decades of the 20th century, there was a rapid development in logic bolstered to a great extent by Polish logicians, especially Alfred Tarski (1901-1983) (see *e.g.* [351]).

Development of computers and their applications stimulated logical research and widened their scope.

When we speak about logic, we generally mean *deductive logic*. It gives us tools designed for deriving true propositions from other true propositions. Deductive reasoning always leads to true conclusions. The theory of deduction has well established generally accepted theoretical foundations. Deductive reasoning is the main tool used in mathematical reasoning and found no application beyond it.

Rough set theory has contributed to some extent to various kinds of deductive reasoning. Particularly, various kinds of logics based on the rough set approach have been investigated, rough set methodology contributed essentially to modal logics, many-valued logic, intuitionistic logic and others (see *e.g.* [6, 7, 53, 54, 57, 70, 69, 146, 147, 161, 160, 162, 185, 187, 188, 190, 213, 214, 246, 247, 259, 260, 261, 262, 263, 264, 265, 360, 359, 361, 362]).

A summary of this research can be found in [245, 38] and interested reader is advised to consult these volumes.

In natural sciences (*e.g.*, in physics) *inductive reasoning* is of primary importance. The characteristic feature of such reasoning is that it does not begin from axioms (expressing general knowledge about the reality) like in deductive logic, but some partial knowledge (examples) about the universe of interest are the starting point of this type of reasoning, which are generalized next and they constitute the knowledge about wider reality than the initial one. In contrast to deductive reasoning, inductive reasoning does not lead to true conclusions but only to probable (possible) ones. Also in contrast to the logic of deduction, the logic of induction does not have uniform, generally accepted, theoretical foundations as yet, although many important and interesting results have been obtained, *for example*, concerning statistical and computational learning and others.

Verification of validity of hypotheses in the logic of induction is based on experiment rather than the formal reasoning of the logic of deduction. Physics is the best illustration of this fact.

The research on inductive logic has a few centuries' long history and outstanding English philosopher John Stuart Mill (1806-1873) is considered its father [150].

The creation of computers and their innovative applications essentially contributed to the rapid growth of interest in inductive reasoning. This domain develops very dynamically thanks to computer science. Machine learning, knowledge discovery, reasoning from data, expert systems and others are examples of new directions in inductive reasoning. It seems that rough set theory is very well suited as a theoretical

basis for inductive reasoning. Basic concepts of this theory fit very well to represent and analyse knowledge acquired from examples, which can be next used as starting point for generalization. Besides, in fact rough set theory has been successfully applied in many domains to find patterns in data (data mining) and acquire knowledge from examples (learning from examples). Thus, rough set theory seems to be another candidate as a mathematical foundation of inductive reasoning [24, 177, 308].

The most interesting from computer science point of view is *common sense* reasoning. We use this kind of reasoning in our everyday life, and examples of such kind of reasoning we face in news papers, radio, TV, in political, economic, debates and discussions.

The starting point to such reasoning is the knowledge possessed by the specific group of people (*common knowledge*) concerning some subject and intuitive methods of deriving conclusions from it. We do not have here possibilities of resolving the dispute by means of methods given by deductive logic (reasoning) or by inductive logic (experiment). So the best known methods of solving the dilemma is voting, negotiations or even war. See *for example*, Gulliver's Travels [341], where the hatred between Tramecksan (High-Heels) and Slamecksan (Low-Heels) or disputes between Big-Endians and Small-Endians could not be resolved without a war.

These methods do not reveal the truth or falsity of the thesis under consideration at all. Of course, such methods are not acceptable in mathematics or physics. Nobody is going to solve by voting, negotiations or declare a war – the truth of Fermat's theorem or Newton's laws.

Reasoning of this kind is the least studied from the theoretical point of view and its structure is not sufficiently understood, in spite of many interesting theoretical research in this domain [62]. The meaning of common sense reasoning, considering its scope and significance for some domains, is fundamental and rough set theory can also play an important role in it but more fundamental research must be done to this end [294].

In particular, the rough truth introduced in [213] and studied, *for example* in [7] seems to be important for investigating commonsense reasoning in the rough set framework.

Let us consider a simple example. In the considered decision system we assume $U = Birds$ is a set of birds that are described by some condition attributes from a set A. The decision attribute is a binary attribute $Files$ with possible values *yes* if the given bird flies and *no*, otherwise. Then, we define (relative to an information system $\mathbb{A} = (U,A)$) the set of abnormal birds by $Ab_A(Birds) = \mathsf{LOW}_A(\{x \in Birds : Flies(x) = no\})$. Hence, we have, $Ab_A(Birds) = Birds - \mathsf{UPP}_A(\{x \in Birds : Flies(x) = yes\})$ and $Birds - Ab_A(Birds) = \mathsf{UPP}_A(\{x \in Birds : Flies(x) = yes\})$. It means that for normal birds it is consistent, with knowledge represented by A, to assume that they can fly, that is, it is possible that they can fly. One can optimize $Ab_A(Birds)$ using A to obtain minimal boundary region in the approximation of $\{x \in Birds : Flies(x) = no\}$.

It is worthwhile to mention that in [48] has been presented an approach combining the rough sets with nonmonotonic reasoning. There are distinguished some basic concepts that can be approximated on the basis of sensor measurements and

more complex concepts that are approximated using so called transducers defined by first-order theories constructed over approximated concepts. Another approach to commonsense reasoning has been developed in a number of papers (see, *e.g.* [24, 177, 199, 255, 294]). The approach is based on an ontological framework for approximation. In this approach, approximations are constructed for concepts and dependencies between the concepts represented in a given ontology expressed, *for example*, in natural language. Still another approach combining rough sets with logic programming is discussed in [365].

To recapitulate, the characteristics of the three above mentioned kinds of reasoning are given below:

1. deductive:
   - reasoning method: axioms and rules of inference;
   - applications: mathematics;
   - theoretical foundations: complete theory;
   - conclusions: true conclusions from true premisses;
   - hypotheses verification: formal proof.
2. inductive:
   - reasoning method: generalization from examples;
   - applications: natural sciences (physics);
   - theoretical foundation: lack of generally accepted theory;
   - conclusions: not true but probable (possible);
   - hypotheses verification - experiment.
3. common sense:
   - reasoning method based on common-sense knowledge with intuitive rules of inference expressed in natural language;
   - applications: every day life, humanities;
   - theoretical foundation: lack of generally accepted theory;
   - conclusions obtained by mixture of deductive and inductive reasoning based on concepts expressed in natural language, *for example*, with application of different inductive strategies for conflict resolution (such as voting, negotiations, cooperation, war) based on human behavioural patterns;
   - hypotheses verification - human behaviour.

There are numerous issues related to approximate reasoning under uncertainty. These issues are discussed in books on granular computing, rough mereology and computational complexity of algorithmic problems related to these issues. For more detail, the reader is referred to the following books [223, 156, 45, 159, 248].

Finally, we would like to stress that still much more work should be done to develop approximate reasoning methods for making progress in development intelligent systems. This idea was very well expressed by Professor Leslie Valiant [33]:

[33] `http://people.seas.harvard.edu/~valiant/researchinterests.htm`

A fundamental question for artificial intelligence is to characterize the computational building blocks that are necessary for cognition. A specific challenge is to build on the success of machine learning so as to cover broader issues in intelligence.... This requires, in particular a reconciliation between two contradictory characteristics – the apparent logical nature of reasoning and the statistical nature of learning.

3.11 Interactive Rough Granular Computing (IRGC)

There are many real-life problems that are still hard to solve using the existing methodologies and technologies. Among such problems are, *for example*, classification and understanding of medical images, control of autonomous systems like unmanned aerial vehicles or robots, and problems related to monitoring or rescue tasks in multiagent systems. All of these problems are closely related to intelligent systems that are more and more widely applied in different real-life projects.

One of the main challenges in developing intelligent systems is discovering methods for approximate reasoning from measurements to perception, that is, deriving from concepts resulting from sensor measurements concepts or expressions enunciated in natural language that express perception understanding.

Nowadays, new emerging computing paradigms are investigated attempting to make progress in solving problems related to this challenge. Further progress depends on a successful cooperation of specialists from different scientific disciplines such as mathematics, computer science, artificial intelligence, biology, physics, chemistry, bioinformatics, medicine, neuroscience, linguistics, psychology, sociology. In particular, different aspects of reasoning from measurements to perception are investigated in psychology [11, 95], neuroscience [244], layered learning [332], mathematics of learning [244], machine learning, pattern recognition [97], data mining [115] and also by researchers working on recently emerged computing paradigms such as computing with words and perception [386], granular computing [199], rough sets, rough-mereology and rough-neural computing [199].

One of the main problems investigated in machine learning, pattern recognition [97] and data mining [115] is concept approximation. It is necessary to induce approximations of concepts (models of concepts) from available experimental data. The data models developed so far in such areas like statistical learning, machine learning, pattern recognition are not satisfactory for approximation of compound concepts resulting in the perception process. Researchers from the different areas have recognized the necessity to work on new methods for concept approximation (see, *e.g.* [35, 364]). The main reason is that these compound concepts are, in a sense, too far from measurements which makes the searching for relevant (for their approximation) features infeasible in a huge space. There are several research directions aiming at overcoming this difficulty. One of them is based on the interdisciplinary research where the results concerning perception in psychology or neuroscience are used to help to deal with compound concepts (see, *e.g.* [97]). There is a great effort in neuroscience towards understanding the hierarchical structures of neural networks in living organisms [56, 244]. Convolutional networks

(ConvNets) which are a biologically inspired trainable architecture that can learn invariant features, were developed (see, *e.g.* [352]). Also mathematicians are recognizing problems of learning as the main problem of the current century [244]. The problems discussed so far are also closely related to complex system modelling. In such systems, again the problem of concept approximation and reasoning about perceptions using concept approximations is one of the challenges nowadays. One should take into account that modelling complex phenomena entails the use of local models (captured by local agents, if one would like to use the multi-agent terminology [103, 387]) that next should be fused [354]. This process involves the negotiations between agents [103] to resolve contradictions and conflicts in local modelling. This kind of modelling will become more and more important in solving complex real-life problems that we are unable to model using traditional analytical approaches. The latter approaches lead to exact models. However, the necessary assumptions used to develop them are causing the resulting solutions to be too far from reality to be accepted. New methods or even a new science should be developed for such modelling [66].

One of the possible solutions in searching for methods for compound concept approximations is the layered learning idea [332]. Inducing concept approximation should be developed hierarchically starting from concepts close to sensor measurements to compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use principles of behaviour on the roads, expressed in natural language, trying to estimate, from recordings (made, *e.g.* by camera and other sensors) of situations on the road, if the current situation on the road is safe or not. To solve such a problem one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. Foundations of such an approach are based on rough set theory [215] and its extension rough mereology [199, 249, 250, 252, 13, 248], both discovered in Poland.

Objects we are dealing with are information granules. Such granules are obtained as the result of information granulation [386]:

> *Information granulation can be viewed as a human way of achieving data compression and it plays a key role in implementation of the strategy of divide-and-conquer in human problem-solving.*

Constructions of information granules should be robust with respect to their input information granule deviations. In this way also a granulation of information granule constructions is considered. As the result, we obtain the so-called AR schemes (AR networks) [199, 249, 250, 252]. AR schemes can be interpreted as complex patterns [115]. Searching methods for such patterns relevant for a given target concept have been developed [199, 13]. Methods for deriving relevant AR schemes are of high computational complexity. The complexity can be substantially reduced by using domain knowledge. In such a case AR schemes are derived along reasoning schemes in natural language that are retrieved from domain knowledge. Developing methods for deriving such AR schemes is one of the main goals of our projects.

Granulation is a computing paradigm, among others like self-reproduction, self-organization, functioning of brain, Darwinian evolution, group behaviour, cell membranes and morphogenesis, that are abstracted from natural phenomena. Granulation is inherent in human thinking and reasoning processes. Granular computing (GrC) provides an information processing framework where computation and operations are performed on information granules, and it is based on the realization that precision is sometimes expensive and not much meaningful in modelling and controlling complex systems. When a problem involves incomplete, uncertain, and vague information, it may be difficult to differentiate distinct elements and one may find it convenient to consider granules for its handling. The structure of granulation can be often defined using methods based on rough sets, fuzzy sets or their combination. In this consortium, rough sets and fuzzy sets work synergistically, often with other soft computing approaches, and use the principle of granular computing. The developed systems exploit the tolerance for imprecision, uncertainty, approximate reasoning and partial truth under soft computing framework and is capable of achieving tractability, robustness and close resemblance with human-like (natural) decision making for pattern recognition in ambiguous situations [292]. Qualitative reasoning requires to develop methods supporting approximate reasoning under uncertainty about non-crisp concepts, often vague concepts. One of the very general scheme of tasks for such qualitative reasoning can be described as follows. From some basic objects (called in different areas as patterns, granules or molecules) it is required to construct (induce) complex objects satisfying a given specification (often, expressed in natural language specification) to a satisfactory degree. For example, in learning concepts from examples we deal with tasks where a partial information about the specification is given by examples and counter examples concerning of classified objects. As examples of such complex objects one can consider classifiers considered in Machine Learning or Data Mining, new medicine against some viruses or behavioural patterns of cell interaction induced from interaction of biochemical processes realized in cells. Over the years we have learned how to solve some of such tasks while many of them are still challenges. One of the reasons is that the discovery process of complex objects relevant for the given specification requires multilevel reasoning with necessity of discovering on each level the relevant structural objects and their properties. The searching space for such structural objects and properties is very huge and this, in particular, shows that fully automatic methods are not feasible using the exiting computing technologies. However, this process can be supported by domain knowledge used which can be used for generating hints in the searching process (see, *e.g.* [13]). This view is consistent with [34] (see, page 3 of Foreword):

> *[...] Tomorrow, I believe, every biologist will use computer to define their research strategy and specific aims, manage their experiments, collect their results, interpret their data, incorporate the findings of others, disseminate their observations, and extend their experimental observations - through exploratory discovery and modelling - in directions completely unanticipated.*

Rough sets, discovered by Zdzisław Pawlak [212], and fuzzy sets, due to Lotfi Zadeh [385], separately and in combination have shown quite strong potential for

supporting the searching process for the relevant complex objects (granules) discussed above (see, *e.g.* [200, 199, 220, 13, 192]). Fuzzy set theory addresses gradualness of knowledge, expressed by the fuzzy membership, whereas rough set theory addresses granularity of knowledge, expressed by the indiscernibility relation.

Computations on granules should be interactive. This requirement is fundamental for modelling of complex systems [67]. For example, in [183] this is expressed by the following sentence:

> *[...] interaction is a critical issue in the understanding of complex systems of any sorts: as such, it has emerged in several well-established scientific areas other than computer science, like biology, physics, social and organizational sciences.*

Interactive Rough Granular Computing (IRGC) is an approach for modelling interactive computations (see, *e.g.* [312]). IRGC are progressing by interactions between granules (structural objects of quite often high-order type) discovered from data and domain knowledge. In particular, interactive information systems (IIS) are dynamic granules used for representing the results of the agent interaction with the environments. IIS can be also applied in modelling more advanced forms of interactions such as hierarchical interactions in layered granular networks or generally in hierarchical modelling. The proposed approach [312, 313] is based on rough sets but it can be combined with other soft computing paradigms such as fuzzy sets or evolutionary computing, and also with machine learning and data mining techniques. The notion of the highly interactive granular system is clarified as the system in which intrastep interactions with the external as well as with the internal environments take place. Two kinds of interactive attributes are distinguished as follows: perception attributes, including sensory ones and action attributes.

The outlined research directions in this section create a step towards understanding the nature of reasoning from measurements to perception. These foundations are crucial for constructing intelligent systems for many real-life projects. The recent progress in this direction based on rough sets and granular computing is reported in [312, 313].

In the following section, we outline three important challenging topics.

3.11.1 Context Inducing and IRGC

Reasoning about context belongs to the main problems investigated in AI for many years (see, *e.g.* [148, 273, 327]). One of the old and still challenging problem in machine learning, pattern recognition and data mining is feature discovery (feature construction, feature extraction) [97]. This problem is related to discovery of structures of objects or contexts in which analyzed objects should be considered. In this section, we discuss an application of information systems for context modelling. The approach is based on fusion of information systems (or decision systems) with constraints. The constraints can be defined by means of relations over sets of attribute values or their Cartesian products. Objects on the next level of modelling are

relational structures over signatures (or sets of signatures) of arguments of fusion operation. In this way, one can obtain as objects on higher level of modelling indiscernibility (similarity) classes of objects, time windows, their clusters, sequences of time windows and their sets. Indiscernibility classes over objects representing sequences of time windows are sets of such sequences and they may represent information about processes.

Let us consider one simple example illustrating this approach elaborated, *for example*, in [311, 312, 313].

In the process of searching for (sub-)optimal approximation spaces, different strategies may be used. Let us consider an example of such strategy presented in [309]. In this example, $DT = (U, A, d)$ denotes a decision system (a given sample of data), where U is a set of objects, A is a set of attributes and d is a decision. We assume that for any object $x \in U$, only partial information equal to the A-signature of x (object signature, for short) is accessible, that is, $Inf_A(x) = \{(a, a(x)) : a \in A\}$. Analogously, for any concept there are only given a partial information about this concept by means of a sample of objects, *for example*, in the form of decision table. One can use object signatures as new objects in a new relational structure $\mathcal{R}$. In this relational structure $\mathcal{R}$ some relations between object signatures are also modeled, *for example*, defined by the similarities of these object signatures (see Figure 3.4).

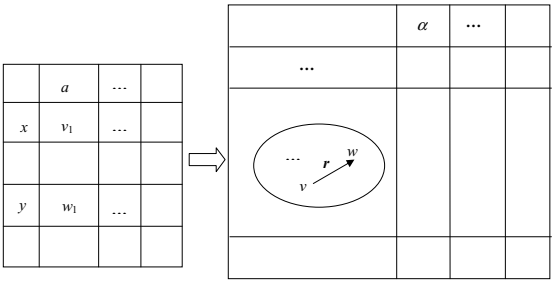


Fig. 3.4. Granulation to tolerance classes. r is a similarity (tolerance) relation defined over signatures of objects.

Discovery of relevant relations between object signatures is an important step in searching for relevant approximation spaces. In this way, a class of relational structures representing perception of objects and their parts is constructed. In the next step, we select a language $\mathcal{L}$ consisting of formulas expressing properties over the defined relational structures and we search for relevant formulas in $\mathcal{L}$. The semantics of formulas (*e.g.* with one free variable) from $\mathcal{L}$ are subsets of object signatures. Note, that each object signature defines a neighborhood of objects from a given sample (*e.g.* decision system DT) and another set on the whole universe of objects being an extension of U. Thus, each formula from $\mathcal{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. Such

families can be used to define new neighbourhoods for a new approximation space by, *for example* taking their unions. In the process of searching for relevant neighbourhoods, we use information encoded in the available sample. More relevant neighbourhoods make it possible to define more relevant approximation spaces (from the point of view of the optimization criterion). Following this scheme, the next level of granulation may be related to clusters of objects (relational structures) for a current level (see Figure 3.5).

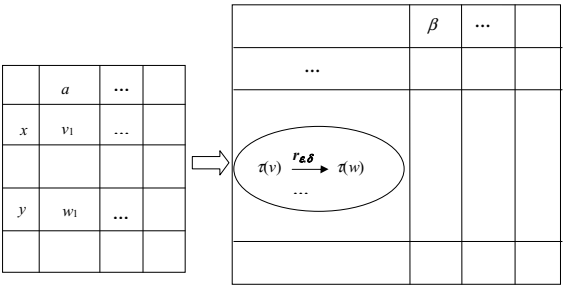


Fig. 3.5. Granulation of tolerance relational structures to clusters of such structures. $r_{\varepsilon,\delta}$ is a relation with parameters ε, δ on similarity (tolerance) classes.

In Figure 3.5, τ denotes a similarity (tolerance) relation on vectors of attribute values, $\tau(v) = \{u : v \ \tau \ u\}$, $\tau(v) \ r_{\varepsilon,\delta} \ \tau(w)$ iff $dist(\tau(v),\tau(w)) \in [\varepsilon - \delta, \varepsilon + \delta]$ and $dist(\tau(v),\tau(w)) = inf\{dist(v',w') : (v',w') \in \tau(v) \times \tau(w)\}$ where $dist$ is a distance function on vectors of attribute values.

One more example is illustrated in Figure 3.6, where the next level of hierarchical modelling is created by defining an information system in which objects are time windows and attributes are (time-related) properties of these windows.

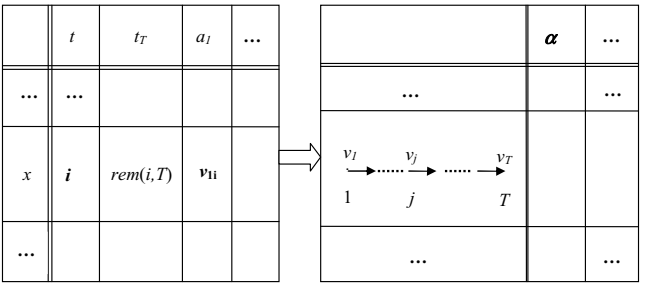


Fig. 3.6. Granulation of time points into time windows. T is the time window length, $v_j = (v_{1j}, \ldots, v_{Tj})$ for $j = 1, \ldots, T$, $rem(i,T)$ is the remainder from division of i by T, α is an attribute defined over time windows.

It is worth mentioning that quite often this searching process is even more sophisticated. For example, one can discover several relational structures (*e.g.* corresponding to different attributes) and formulas over such structures defining different families of neighbourhoods from the original approximation space. As a next step, such families of neighborhoods can be merged into neighbourhoods in a new, higher degree approximation space.

The proposed approach is making it possible to construct information systems (or decision system) on a given level of hierarchical modelling from information systems from lower level(s) by using some constraints in joining objects from underlying information systems. In this way, structural objects can be modelled and their properties can be expressed in constructed information systems by selecting relevant attributes. These attributes are defined with use of a language that makes use of attributes of systems from the lower hierarchical level as well as relations used to define constraints. In some sense, the objects on the next level of hierarchical modelling are defined using the syntax from the lover level of the hierarchy. Domain knowledge is used to aid the discovery of relevant attributes (features) on each level of hierarchy. This domain knowledge can be provided, *for example*, by concept ontology together with samples of objects illustrating concepts from this ontology. Such knowledge is making it feasible to search for relevant attributes (features) on different levels of hierarchical modelling.

In Figure 3.7 we symbolically illustrate the transfer of knowledge in a particular application. It is a depiction of how the knowledge about outliers in handwritten digit recognition is transferred from expert to a software system. We call this process *knowledge elicitation* [179, 180, 181]. Observe, that the explanations given by expert(s) are expressed using a subset of natural language limited by using concepts from provided ontology only. Concepts from higher levels of ontology are gradually approximated by the system from concepts on lower levels.

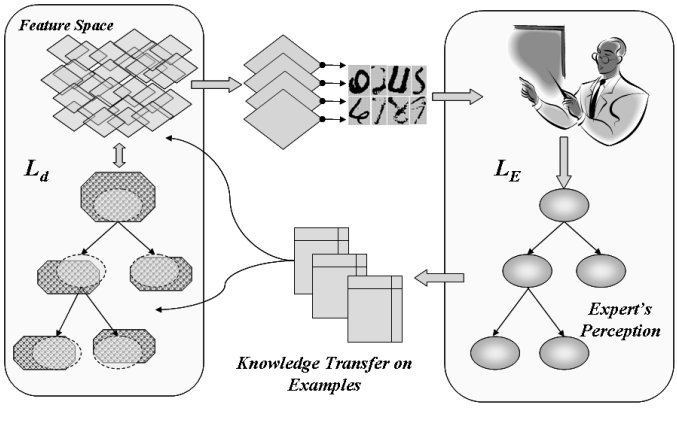


Fig. 3.7. Expert's knowledge elicitation.

This kind of approach is typical for hierarchical modelling [25, 17, 24, 177, 13, 14]. This is, in particular, the case when we search for a relevant approximation space for objects composed from parts for which some approximation spaces, relevant to components, have already been found. We find that hierarchical modelling is required for approximation of complex vague concepts, as in [181, 244].

3.11.2 Process Mining and IRGC

The rapid expansion of the Internet has resulted not only in the ever-growing amount of data therein stored, but also in the burgeoning complexity of the concepts and phenomena pertaining to those data. This issue has been vividly compared in [60] with the advances in human mobility from the period of walking afoot to the era of jet travel. These essential changes in data have brought new challenges to the development of new data mining methods, especially that the treatment of these data increasingly involves complex processes that elude classic modelling paradigms. Types of datasets currently regarded "hot", like biomedical, financial or net user behaviour data are just a few examples. Mining such temporal or complex data streams is on the agenda of many research centres and companies worldwide (see, *for example*, [1, 269]). In the data mining community, there is a rapidly growing interest in developing methods for process mining, *e.g.*, for discovery of structures of temporal processes from observations (recorded data). Works on process mining, *for example*, [33, 149, 358, 376, 363] have recently been undertaken by many renowned centres worldwide[34]. This research is also related to functional data analysis (cf. [258]), cognitive networks (cf. [208]) and dynamical system modelling in biology (cf. [58]).

Let us consider an illustrative example explaining motivation for discovery of process models from data.

This problem is illustrated in Figure 3.8. It is assumed that from granules G, G_1, G_2 representing the sets of the paths of the processes, their models in the form of Petri nets PN, PN_1, PN_2, respectively, were induced. Then, the structure of interaction between PN_1 and PN_2 can be described by an operation transforming PN_1, PN_2 into PN.

The discovery of relevant attributes on each level of the hierarchy can be supported by domain knowledge provided, *for example*, by concept ontology together with the illustration of concepts by means of the samples of objects taken from this concepts and their complements [13]. Such application of domain knowledge often taken from human experts serves as another example of the interaction of a system (classifier) with its environment. Additionally, such support of relevant attributes discovery on given level of the hierarchy, as well as on other levels, can be found using different ontologies. These ontologies can be described by different sets of

[34] http://www.isle.org/∼langley/, http://soc.web.cse.unsw.edu.au/bibliography/discovery/index.html, http://www.processmining.org/

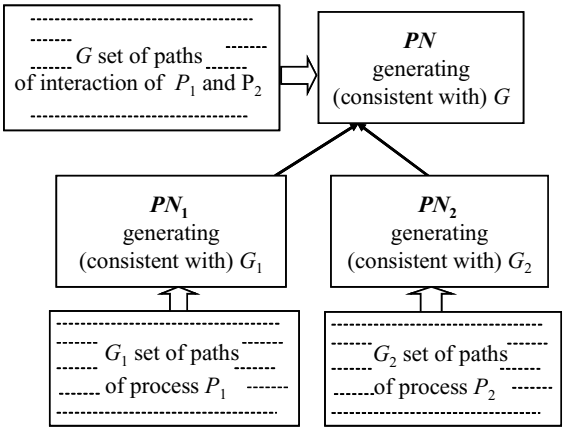


Fig. 3.8. Discovery of interaction structure

formulas and possibly by different logics. Thus, the description of such discovery of relevant attributes in interaction, as well as its support give a good reason for applying fibring logics methods [61]. Note that in the hierarchical modelling of relevant complex patterns also top-down interactions of the higher levels of the hierarchy with the lower levels should be considered, *for example*, if the patterns constructed on higher levels are not relevant for the target task, the top-down interaction should inform lower levels about the necessity of searching for new patterns.

There are numerous papers based on the rough set approach on discovery of concurrent processes from data (see, *e.g.* [305, 304, 306, 334, 239, 335, 240, 201, 337, 202, 203, 204, 205, 340, 339, 346, 338, 206, 154, 207, 157, 155, 158, 336, 170, 45]). In [174, 170] was outlined an approach to discovery of processes from data and domain knowledge which is based on RGC philosophy. This research was initiated by the idea of Professor Zdzisław Pawlak presented in [216], where data tables are treated as partial specifications of concurrent processes. Rows in a given data table are examples of global states and columns represent local processes. One of the solutions presented in the above papers was based on decomposition of data tables into modules defined by reducts of data tables. The modules are linked by constraints defined by rules extracted from data. In another approach, first from a given data table decision rules are extracted (*e.g.*, a set of minimal decision rules) and such a set of decision rules is used as knowledge encoded in the data table or theory defined by data table. Next, the set of all global states is defined as equal to the maximal set of objects (global states) consistent with the theory. There were proposed methods for automatic generation from a given data table a (colored) Petri net with the reachability set equal to the maximal consistent set of states consistent with the theory generated from the data table. The reader is referred to the Web page `http://rsds.univ.rzeszow.pl` for information on the developed software (ROSECON) for inducing Petri nets from data tables.

An important role in discovering Petri nets play the inhibitory rules (see, *e.g.* [45]). The reader interested in complexity results related to such rules as well as to consistent extensions of information systems is referred to, *for example*, [154, 157, 155, 45, 158].

Here, we would like to formulate some challenges related to discovery of concurrent processes from data tables occurring in hierarchical modelling by using IRGC. On higher levels of hierarchy, the structure of objects becomes complex, *e.g.*, indiscernibility classes of data tables considered on higher level of the hierarchy can be equal to sets of paths of structural states. The theories of such data tables are much more complex than considered before. The rules in such a theory discovered from data may require extension to (spatio-)temporal decision rules or temporal association rules or even more complex rules defined by different temporal logics. The challenges are related to discovery of relevant rules and theories over such rules as well as to inducing, for example, Petri nets consistent with theories defined by such constraints.

3.11.3 Perception-Based Computing and IRGC

Perception-Based Computing (PBC) methods are needed to face problems of data mining (DM) and knowledge discovery in databases (KDD) with dynamically evolving complex data (*e.g.* stream data sources, sensory data). Another challenge, making PBC methods indispensable, is a growth of the size and complexity of data sources (*e.g.* Web sources, neuro-imaging data, data from network interactions) in open environments. These challenges, in particular, discovery of complex concepts such as behavioural patterns, hardly can be met by classical methods [244]. They can be met by KDD systems which dialogue with experts or users during the discovery process[364] or by adaptive learning systems changing themselves during the learning process as the response to evolving data.

Another area where PBC methods are needed is a multi-agent systems field. Behaviour steering and coordination of multi-agent coalitions acting and cooperating in open, unpredictable environments call for interactive algorithms [68], that is, algorithms interacting with the environment during performing particular steps of computations or changing themselves during the process of computation. Next challenge of this type comes from human – robot interaction. The problem of human control over autonomously coordinating swarms of robots is the central challenge in this field which should be solved before human == robot teams can be taken out of laboratories and put to practical use.

Coordination and control are essentially perception based thus PBC methods are indispensable for designing and behaviour description of cognitive systems and for understanding interactions in layered granular networks [223], where granules can be interpreted both as data patterns and agents (*e.g.* robots or movable sensors). Granules in such networks, which are additionally self-organizing, can be also understood as cores in pertinent multi-core computing engines in structurally and

run-time reconfigurable hardware, what makes PBCs useful in computer engineering as well as an essential part of cognitive informatics.

Current works are aimed at developing methods based on the generalized information systems (a special kind of data tables) and the rough set approach for representing partial information on interactions in layered granular networks [106, 108, 107], [312, 313]. The idea of the representation of interactions using information systems has some roots in such approaches as rough sets introduced by Zdzisław Pawlak [212], the information flow by Jon Barwise [12] or Chu spaces [10], http://chu.stanford.edu/. Information systems are used to represent granules of different complexity and the interactions among them [312, 107]. Rough sets are used for vague concept approximation [222], for example, in the approximation of ontologies given by experts (see, *e.g.* [13]).

Perception-based computing provides capability to compute and reason with perception-based information as humans do to perform a wide variety of physical and mental tasks without any measurement and computation. Reflecting the finite ability of the sensory organs and (finally the brain) to resolve details, perceptions are inherently granular. Boundaries of perceived granules (*e.g.* classes) are unsharp and the values of the attributes they can take are granulated. In general, perception may be considered as understanding of sensory information. This point of view is, *for example*, presented in Computing with Words and Perception [386] which

> *derives from the fact that it opens the door to computation and reasoning with information which is perception – rather than measurement-based. Perceptions play a key role in human cognition, and underlie the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis and summarizing a story*.

The need for perception based computing appears, for example, in problems of analysis of complex processes that result from the interaction of many component processes and from control over such process. A component process control is aimed at achieving the desired patterns of the system behaviors. This task is a challenge for areas such as multi-agent systems or autonomous systems [348, 282, 141]. Perceived properties of complex processes are often complex vague concepts, about which only partial information is available. Also information about the satisfiability of such concepts determines activating complex actions. It is worth noting that actions can be initiated at different levels of the hierarchy of concepts from a given ontology and that a prediction of action activation at higher levels of the hierarchy is usually conditioned by the perception of properties depending on the history of computations in which information about actions conducted also on lower levels of the hierarchy and their results is stored. Discovery of search strategies for new essential properties at higher levels of hierarchy becomes a challenge and is crucial for understanding of perception. The values of these attributes depend on the history of computing (with registered information about the actions performed on the actual and the lower levels and their results). These new features determine the perception of satisfiability degrees of complex concepts mentioned above conditioning execution of actions on the considered level of hierarchy. The difficulties of analysis

and synthesis of perceptual computations follow from the nature of interactive computations, in which it becomes necessary to take into account interactions between processes during performed steps of computing (called intrastep interactions [94]). These difficulties follow also from partial information about component processes, from possible interactions between them, and also from requirements on avoidance of central control.

There are several critical issue for making progress in perception understanding and modelling. Among them is the nature of objects on which are performed computations leading to perception. We propose to use granules for modelling such objects. The computations on granules are realized through interactions.

Note also that the fusion of information may lead to new information systems with structural objects [312, 313, 311] or to nets of information systems linked by different constraints. For example, a family of parameterized sensors may model a situation in which the sensors are enabled by the judgment module for recording features of video at different moments of time in probing the environment. This makes it possible to collect the necessary features of the environment for an activating of the relevant higher level action. Parameters may be related, *for example*, to positions of moving camera. This is closely related to the approach to perception presented in [182] (page 1) (see also Figure 3.9):

> *[...] perceiving is a way of acting. Perception is not something that happens to us, or in us. It is something we do. Think of blind person tap-tapping his or her way around a cluttered space, perceiving the space by touch, not all at once, but through time, by skillful probing and movement. This is, or at least ought to be, our paradigm of what perceiving is. The world makes itself available to the perceiver through physical movement and interaction.*

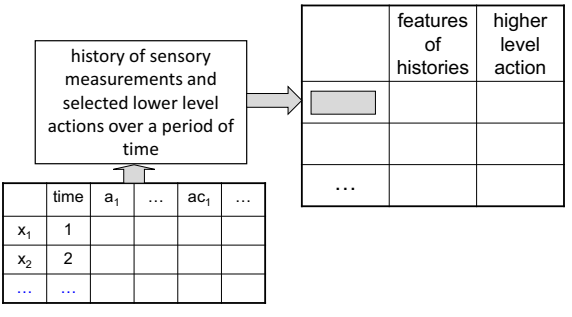


Fig. 3.9. Action in perception

The last example suggests that the sensory attributes may be fused using some parameters such as time of enabling or position of sensors. Certainly, for performing more compound actions it is necessary to use a net of such parameterized sensors in which sensory attributes are linked with relevant constraints [182]. Hierarchical modelling may also lead to nets of information systems constructed over information systems corresponding to sensory attributes. Nodes in these networks may be

linked using different information such as behavioural patterns or local theories induced from information systems in nodes as well as their changes when information systems are updated. In the former case, the reader may recognize some analogy to theory of information flow [12].

We proposed to build foundations for Perception based Computing (PBC) on the basis of Interactive Granular Computing (IGC), in particular on Interactive Rough Granular Computing (IRGC). A step towards this goal is presented in [312, 313]. PBC can be considered in a more general framework of Wisdom Technology (Wistech) [106, 108, 107] based on a metaequation

$$wisdom = knowledge + adaptive\ judgment + interactions. \tag{3.50}$$

In the above metaequation there is mentioned a special kind of reasoning called as *adaptive judgment*. There are many important issues belonging to adaptive judgment such as searching for relevant approximation spaces including inducing new features, feature selection, rule induction, discovery of measures of inclusion and strategies for conflict resolution, adaptation of measures based on the minimum description length, adaptive reasoning about changes, perception (action and sensory) attributes selection, adaptation of quality measures during computations performed by agents, adaptation of object structures, adaptation of strategies for knowledge representation and interaction with knowledge bases, ontology acquisition and approximation, discovery of language for cooperation or competition, and adaptive strategies for language evolution. In general, adaptive judgment is a mixture of deductive and inductive reasoning methods for reasoning about interactive granular computations and on controlling such computations by adaptive strategies for achieving the target goals. The mentioned mixture of deductive and inductive reasoning creates many challenges. This is closely related to opinion expressed by Leslie Valiant `http://people.seas.harvard.edu/~valiant/researchinterests.htm`

> *A fundamental question for artificial intelligence is to characterize the computational building blocks that are necessary for cognition. A specific challenge is to build on the success of machine learning so as to cover broader issues in intelligence. This requires, in particular a reconciliation between two contradictory characteristics – the apparent logical nature of reasoning and the statistical nature of learning.*

3.12 Conclusions

In the chapter, we have discussed some basic issues and methods related to rough sets together with some generalizations, including those related to relationships of rough sets with inductive reasoning. We have also listed some research directions based on interactive rough granular computing. For more detail the reader is referred to the literature cited at the beginning of this chapter (see also `http://rsds.wsiz.rzeszow.pl`).

We are observing a growing research interest in the foundations and applications of rough sets.

Relationships between rough sets and other approaches have been established as well as a wide range of hybrid systems have been developed. In particular, its relationships to fuzzy set theory, the theory of evidence, Boolean reasoning methods, statistical methods and decision theory have been clarified and seem now to be thoroughly understood. There are reports on many hybrid methods obtained by combining the rough set approach with other approaches such as fuzzy sets, neural networks, genetic algorithms, principal component analysis, singular value decomposition or vector support machines.

Rough sets are now linked with decision system used for modelling and analysis of complex systems, fuzzy sets, neural networks, evolutionary computing, data mining and knowledge discovery, pattern recognition, machine learning, data mining and approximate reasoning, multicriteria decision making. In particular, rough sets are used in probabilistic reasoning, granular computing, intelligent control, intelligent agent modelling, identification of autonomous systems and process specification.

A wide range of applications of methods based on rough set theory alone or in combination with other approaches have been discovered in many areas including: acoustics, bioinformatics, business and finance, chemistry, computer engineering (*e.g.* data compression, digital image processing, digital signal processing, parallel and distributed computer systems, sensor fusion, fractal engineering), decision analysis and systems, economics, electrical engineering (*e.g.* control, signal analysis, power systems), environmental studies, digital image processing, informatics, medicine, molecular biology, musicology, neurology, robotics, social science, software engineering, spatial visualization, Web engineering, and Web mining.

Many important research topics in rough set theory such as various logics related to rough sets and many advanced algebraic properties of rough sets were only mentioned in the chapter. The reader can find details in the books, articles and journals cited in this chapter.

Acknowledgements. This research has been supported by the grant from KAUST, the grant NN516 077837 from the Ministry of Science and Higher Education of the Republic of Poland and by the National Centre for Research and Development (NCBiR) under grant SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

References

1. Aggarwal, C.: Data Streams: Models and Algorithms. Springer, Berlin (2007)
2. Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.): RSCTC 2002. LNCS (LNAI), vol. 2475. Springer, Heidelberg (2002)
3. An, A., Huang, Y., Huang, X., Cercone, N.: Feature selection with rough sets for web page classification. In: Peters et al. [231], pp. 1–13

4. An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.): RSFDGrC 2007. LNCS (LNAI), vol. 4482. Springer, Heidelberg (2007)
5. Ariew, R., Garber, D. (eds.): Leibniz, G. W., Philosophical Essays. Hackett Publishing Company, Indianapolis (1989)
6. Balbiani, P., Vakarelov, D.: A modal logic for indiscernibility and complementarity in information systems. Fundamenta Informaticae 50(3-4), 243–263 (2002)
7. Banerjee, M., Chakraborty, M.: Logic for rough truth. Fundamenta Informaticae 71(2-3), 139–151 (2006)
8. Banerjee, M., Pal, S.K.: Roughness of a fuzzy set. Information Sciences 93(3-4), 235–246 (1996)
9. Bargiela, A., Pedrycz, W. (eds.): Granular Computing: An Introduction. Kluwer Academic Publishers (2003)
10. Barr, B.: *-Autonomous categories, Lecture Notes in Mathematics, vol. 752. Springer (1979)
11. Barsalou, L.W.: Perceptual symbol systems. Behavioral and Brain Sciences 22, 577–660 (1999)
12. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press (1997)
13. Bazan, J.: Hierarchical classifiers for complex spatio-temporal concepts. In: Peters et al. [236], pp. 474–750
14. Bazan, J.: Rough sets and granular computing in behavioral pattern identification and planning. In: Pedrycz et al. [223], pp. 777–822
15. Bazan, J., Latkowski, R., Szczuka, M.: DIXER - Distributed executor for rough set exploration system. In: Ślęzak et al. [321], pp. 362–371
16. Bazan, J., Nguyen, H.S., Nguyen, S.H., Synak, P., Wróblewski, J.: Rough set algorithms in classification problems. In: Polkowski et al. [249], pp. 49–88
17. Bazan, J., Skowron, A.: On-line elimination of non-relevant parts of complex objects in behavioral pattern identification. In: Pal et al. [193], pp. 720–725
18. Bazan, J.G., Skowron, A., Ślęzak, D., Wróblewski, J.: Searching for the Complex Decision Reducts: The Case Study of the Survival Analysis. In: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (eds.) ISMIS 2003. LNCS (LNAI), vol. 2871, pp. 160–168. Springer, Heidelberg (2003)
19. Bazan, J., Szczuka, M., Wojna, M., Wojnarski, M.: On the evolution of rough set exploration system. In: Tsumoto et al. [356], pp. 592–601
20. Bazan, J.G.: A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In: Polkowski and Skowron [252], pp. 321–365
21. Bazan, J.G., Nguyen, H.S., Peters, J.F., Skowron, A., Szczuka, M.: Rough set approach to pattern extraction from classifiers. In: Skowron and Szczuka [310], pp. 20–29, www.elsevier.nl/locate/entcs/volume82.html
22. Bazan, J.G., Nguyen, H.S., Skowron, A., Szczuka, M.: A view on rough set concept approximation. In: Wang et al. [367], pp. 181–188
23. Bazan, J.G., Nguyen, H.S., Szczuka, M.S.: A view on rough set concept approximations. Fundamenta Informaticae 59, 107–118 (2004)
24. Bazan, J.G., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: Ślęzak et al. [321], pp. 688–697
25. Bazan, J.G., Skowron, A.: Classifiers based on approximate reasoning schemes. In: Dunin-Kęplicz et al. [52], pp. 191–202
26. Bazan, J.G., Skowron, A., Swiniarski, R.: Rough sets and vague concept approximation: From sample approximation to adaptive learning. In: Peters and Skowron [228], pp. 39–62

27. Bazan, J.G., Szczuka, M.: RSES and RSESlib - a collection of tools for rough set computations. In: Ziarko and Yao [394], pp. 106–113
28. Behnke, S.: Hierarchical Neural Networks for Image Interpretation. LNCS, vol. 2766. Springer, Heidelberg (2003)
29. Bello, R., Falcón, R., Pedrycz, W.: Computing: At the Junction of Rough Sets and Fuzzy Sets. STUDFUZZ, vol. 234. Springer, Heidelberg (2010)
30. Blake, A.: Canonical expressions in Boolean algebra. Dissertation, Dept. of Mathematics, University of Chicago (1937); University of Chicago Libraries (1938)
31. Boole, G.: The Mathematical Analysis of Logic. G. Bell, London (1847); reprinted by Philosophical Library, New York (1948)
32. Boole, G.: An Investigation of the Laws of Thought. Walton, London (1854); reprinted by Dover Books, New York (1954)
33. Borrett, S.R., Bridewell, W., Langely, P., Arrigo, K.R.: A method for representing and developing process models. Ecological Complexity 4, 1–12 (2007)
34. Bower, J.M., Bolouri, H. (eds.): Computational Modelling of Genetic and Biochemical Networks. MIT Press (2001)
35. Breiman, L.: Statistical modelling: The two cultures. Statistical Science 16(3), 199–231 (2001)
36. Brown, F.: Boolean Reasoning. Kluwer Academic Publishers, Dordrecht (1990)
37. Cercone, N., Skowron, A., Zhong, N.: (Special issue), Computational Intelligence: An International Journal 17(3) (2001)
38. Chakraborty, M., Pagliani, P.: A Geometry of Approximation: Rough Set Theory: Logic, Algebra and Topology of Conceptual Patterns. Springer, Heidelberg (2008)
39. Chan, C.-C., Grzymała-Busse, J.W., Ziarko, W.P. (eds.): RSCTC 2008. LNCS (LNAI), vol. 5306. Springer, Heidelberg (2008)
40. Chikalov, I., Lozin, V., Lozina, I., Moshkov, M., Nguyen, H.S., Skowron, A., Zielosko, B.: Three Approaches to Data Analysis. Test Theory, Rough Sets and Logical Analysis of Data. Springer (to appear, 2012)
41. Chmielewski, M.R., Grzymała-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning 15(4), 319–331 (1996)
42. Choubey, S.K., Deogun, J.S., Raghavan, V.V., Sever, H.: A comparison of feature selection algorithms in the context of rough classifiers. In: Petry, F. (ed.) International Conference on Fuzzy Systems (FUZZ-IEEE 1996), New Orleans, LA, September 8-11, vol. 2, pp. 1122–1128. IEEE Service Center, Pistcataway (1996)
43. Cios, K., Pedrycz, W., Swiniarski, R.: Data Mining Methods for Knowledge Discovery. Kluwer, Norwell (1998)
44. Ciucci, D., Yao, Y.Y.: Special issue on Advances in Rough Set Theory, Fundamenta Informaticae 108(3-4) (2011)
45. Delimata, P., Moshkov, M.J., Skowron, A., Suraj, Z.: Inhibitory Rules in Data Analysis: A Rough Set Approach. SCI, vol. 163. Springer, Heidelberg (2009)
46. Demri, S., Orłowska, E. (eds.): Incomplete Information: Structure, Inference, Complexity. Monographs in Theoretical Cpmputer Sience. Springer, Heidelberg (2002)
47. Deogun, J., Raghavan, V.V., Sarkar, A., Sever, H.: Data mining: Trends in research and development. In: Lin and Cercone [135], pp. 9–46
48. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Engineering: A Rough Set Approach. SCI, vol. 202. Springer, Heidelberg (2006)
49. Dubois, D., Prade, H.: Foreword. In: Rough Sets: Theoretical Aspects of Reasoning about Data [215]

50. Dubois, V., Quafafou, M.: Concept learning with approximation: Rough version spaces. In: Alpigini et al. [2], pp. 239–246
51. Duda, R., Hart, P., Stork, R.: Pattern Classification. John Wiley & Sons, New York (2002)
52. Dunin-Kęplicz, B., Jankowski, A., Skowron, A., Szczuka, M. (eds.): Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS 2004). Advances in Soft Computing. Springer, Heidelberg (2005)
53. Düntsch, I.: A logic for rough sets. Theoretical Computer Science 179, 427–436 (1997)
54. Düntsch, I., Gediga, G.: Rough set data analysis. In: Encyclopedia of Computer Science and Technology, vol. 43, pp. 281–301. Marcel Dekker (2000)
55. Düntsch, I., Gediga, G.: Rough set data analysis: A road to non-invasive knowledge discovery. Methodos Publishers, Bangor (2000)
56. Fahle, M., Poggio, T.: Perceptual Learning. MIT Press, Cambridge (2002)
57. Fan, T.F., Liau, C.J., Yao, Y.: On modal and fuzzy decision logics based on rough set theory. Fundamenta Informaticae 52(4), 323–344 (2002)
58. Feng, J., Jost, J., Minping, Q. (eds.): Network: From Biology to Theory. Springer, Berlin (2007)
59. Frege, G.: Grundgesetzen der Arithmetik, vol. 2. Verlag von Hermann Pohle, Jena (1903)
60. Friedman, J.H.: Data mining and statistics. What's the connection (keynote address). In: Scott, D. (ed.) Proceedings of the 29th Symposium on the Interface: Computing Science and Statistics, Huston, Texas, May 14-17, University of Huston, Huston (1997)
61. Gabbay, D. (ed.): Fibring Logics. Oxford University Press (1998)
62. Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.): Handbook of Logic in Artificial Intelligence and Logic Programming. Nonmonotonic Reasoning and Uncertain Reasoning, vol. 3. Calderon Press, Oxford (1994)
63. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book. Prentice Hall, Upper Saddle River (2002)
64. Gediga, G., Düntsch, I.: Rough approximation quality revisited. Artificial Intelligence 132, 219–234 (2001)
65. Gediga, G., Düntsch, I.: On model evaluation, indices of importance, and interaction values in rough set analysis. In: Pal et al. [199], pp. 251–276
66. Gell-Mann, M.: The Quark and the Jaguar - Adventures in the Simple and the Complex. Brown and Co., London (1994)
67. Goldin, D., Smolka, S., Wegner, P. (eds.): Interactive Computation: The New Paradigm. Springer (2006)
68. Goldin, D., Wegner, P.: Principles of interactive computation. In: Goldin et al. [67], pp. 25–37
69. Gomolińska, A.: A graded meaning of formulas in approximation spaces. Fundamenta Informaticae 60(1-4), 159–172 (2004)
70. Gomolińska, A.: Rough validity, confidence, and coverage of rules in approximation spaces. In: Peters and Skowron [226], pp. 57–81
71. Góra, G., Wojna, A.G.: RIONA: A new classification system combining rule induction and instance-based learning. Fundamenta Informaticae 51(4), 369–390 (2002)
72. Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.): RSCTC 2006. LNCS (LNAI), vol. 4259. Springer, Heidelberg (2006)
73. Greco, S., Inuiguchi, M., Słowiński, R.: Fuzzy rough sets and multiple-premise gradual decision rules. International Journal of Approximate Reasoning 41(2), 179–211 (2006)

74. Greco, S., Kadzinski, M., Słowiński, R.: Selection of a representative value function in robust multiple criteria sorting. Computers & OR 38(11), 1620–1637 (2011)
75. Greco, S., Matarazzo, B., Słowiński, R.: Dealing with missing data in rough set analysis of multi-attribute and multi-criteria decision problems. In: Zanakis, S., Doukidis, G., Zopounidis, C. (eds.) Decision Making: Recent Developments and Worldwide Applications, pp. 295–316. Kluwer Academic Publishers, Boston (2000)
76. Greco, S., Matarazzo, B., Słowiński, R.: Rough set theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
77. Greco, S., Matarazzo, B., Słowiński, R.: Data mining tasks and methods: Classification: multicriteria classification. In: Kloesgen, W., Żytkow, J. (eds.) Handbook of KDD, pp. 318–328. Oxford University Press, Oxford (2002)
78. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based rough set approach to knowledge discovery (I) - General perspective, (II) - Extensions and applications. In: Zhong and Liu [387], pp. 513–552, 553–612
79. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based rough set approach as a proper way of handling graduality in rough set theory. In: Peters et al. [235], pp. 36–52
80. Greco, S., Matarazzo, B., Słowiński, R.: Granular computing and data mining for ordered data: The dominance-based rough set approach. In: Encyclopedia of Complexity and Systems Science, pp. 4283–4305. Springer, Heidelberg (2009)
81. Greco, S., Matarazzo, B., Słowiński, R.: A summary and update of "Granular computing and data mining for ordered data: The dominance-based rough set approach". In: Hu, X., Lin, T.Y., Raghavan, V.V., Grzymała-Busse, J.W., Liu, Q., Broder, A.Z. (eds.) 2010 IEEE International Conference on Granular Computing, GrC 2010, San Jose, California, USA, August 14-16, pp. 20–21. IEEE Computer Society (2010)
82. Greco, S., Słowiński, R., Stefanowski, J., Zurawski, M.: Incremental versus non-incremental rule induction for multicriteria classification. In: Peters et al. [231], pp. 54–62
83. Grzymała-Busse, J.W.: Managing Uncertainty in Expert Systems. Kluwer Academic Publishers, Norwell (1990)
84. Grzymała-Busse, J.W.: LERS – A system for learning from examples based on rough sets. In: Słowiński [324], pp. 3–18
85. Grzymała-Busse, J.W.: Selected algorithms of machine learning from examples. Fundamenta Informaticae 18, 193–207 (1993)
86. Grzymała-Busse, J.W.: Classification of unseen examples under uncertainty. Fundamenta Informaticae 30(3-4), 255–267 (1997)
87. Grzymała-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31(1), 27–39 (1997)
88. Grzymała-Busse, J.W.: Three strategies to rule induction from data with numerical attributes. In: Peters et al. [231], pp. 54–62
89. Grzymała-Busse, J.W.: LERS - A data mining system. In: Maimon and Rokach [143], pp. 1347–1351
90. Grzymała-Busse, J.W.: Rule induction. In: Maimon and Rokach [143], pp. 277–294
91. Grzymała-Busse, J.W.: Generalized parameterized approximations. In: Yao et al. [378], pp. 136–145
92. Grzymała-Busse, J.W., Grzymała-Busse, W.J.: Handling missing attribute values. In: Maimon and Rokach [143], pp. 37–57
93. Grzymała-Busse, J.W., Ziarko, W.: Data mining and rough set theory. Communications of the ACM 43, 108–109 (2000)
94. Gurevich, Y.: Interactive algorithms 2005. In: Goldin et al. [67], pp. 165–181

95. Harnad, S.: Categorical Perception: The Groundwork of Cognition. Cambridge University Press, New York (1987)
96. Hassanien, A.E., Suraj, Z., Ślęzak, D., Lingras, P. (eds.): Rough Computing: Theories, Technologies and Applications. IGI Global, Hershey (2008)
97. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
98. Herbert, J., Yao, J.T.: Time-series data analysis with rough sets. In: Proceedings of the 4th International Conference on Computational Intelligence in Economics and Finance (CIEF 2005), Salt Lake City, UT, July 21-26, pp. 908–911 (2005)
99. Hirano, S., Inuiguchi, M., Tsumoto, S. (eds.): Proceedings of International Workshop on Rough Set Theory and Granular Computing (RSTGC 2001), Matsue, Shimane, Japan, May 20-22. Bulletin of the International Rough Set Society, vol. 5(1-2). International Rough Set Society, Matsue (2001)
100. Hu, X., Cercone, N.: Learning in relational databases: A rough set approach. Computational Intelligence: An International Journal 11(2), 323–338 (1995)
101. Hu, X., Cercone, N.: Data mining via discretization, generalization and rough set feature selection. Knowledge and Information Systems: An International Journal 1(1), 33–60 (1999)
102. Hu, X., Cercone, N.: Discovering maximal generalized decision rules through horizontal and vertical data reduction. Computational Intelligence: An International Journal 17(4), 685–702 (2001)
103. Huhns, M.N., Singh, M.P.: Readings in Agents. Morgan Kaufmann, San Mateo (1998)
104. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. STUDFUZZ, vol. 125. Springer, Heidelberg (2003)
105. Jain, R., Abraham, A.: Special issue on Hybrid Intelligence using rough sets, International Journal of Hybrid Intelligent Systems 2 (2005)
106. Jankowski, A., Skowron, A.: A wistech paradigm for intelligent systems. In: Peters et al. [232], pp. 94–132
107. Jankowski, A., Skowron, A.: Logic for artificial intelligence: The Rasiowa - Pawlak school perspective. In: Ehrenfeucht, A., Marek, V., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 106–143. IOS Press, Amsterdam (2008)
108. Jankowski, A., Skowron, A.: Wisdom Technology: A Rough-Granular Approach. In: Marciniak, M., Mykowiecka, A. (eds.) Bolc Festschrift. LNCS, vol. 5070, pp. 3–41. Springer, Heidelberg (2009)
109. Jensen, R., Shen, Q.: Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. IEEE Press Series on Cmputational Intelligence. IEEE Press and John Wiley & Sons, Hoboken, NJ (2008)
110. Jian, L., Liu, S., Lin, Y.: Hybrid Rough Sets and Applications in Uncertain Decision-Making (Systems Evaluation, Prediction, and Decision-Making. CRC Press, Boca Raton (2010)
111. Keefe, R.: Theories of Vagueness. Cambridge Studies in Philosophy, Cambridge (2000)
112. Keefe, R., Smith, P.: Vagueness: A Reader. MIT Press, Massachusetts (1997)
113. Kim, D.: Data classification based on tolerant rough set. Pattern Recognition 34(8), 1613–1624 (2001)
114. Kim, D., Bang, S.Y.: A handwritten numeral character classification using tolerant rough set. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(9), 923–937 (2000)
115. Kloesgen, W., Żytkow, J. (eds.): Handbook of Knowledge Discovery and Data Mining. Oxford University Press, Oxford (2002)

116. Komorowski, J., Øhrn, A., Skowron, A.: Rosetta and other software systems for rough sets. In: Klösgen, W., Żytkow, J. (eds.) Handbook of Data Mining and Knowledge Discovery, pp. 554–559. Oxford University Press (2000)
117. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: Pal and Skowron [200], pp. 3–98
118. Kostek, B.: Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Physical Acoustics. STUDFUZZ, vol. 31. Physica-Verlag, Heidelberg (1999)
119. Kostek, B.: Perception-Based Data Processing in Acoustics: Applications to Music Information Retrieval and Psychophysiology of Hearing. SCI, vol. 3. Springer, Heidelberg (2005)
120. Kotlowski, W., Dembczynski, K., Greco, S., Słowiński, R.: Stochastic dominance-based rough set model for ordinal classification. Information Sciences 178(21), 4019–4037 (2008)
121. Kryszkiewicz, M., Rybiński, H.: Computation of reducts of composed information systems. Fundamenta Informaticae 27(2-3), 183–195 (1996)
122. Kryszkiewicz, M., Cichoń, K.: Towards scalable algorithms for discovering rough set reducts. In: Peters and Skowron [233], pp. 120–143
123. Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.): RSEISP 2007. LNCS (LNAI), vol. 4585. Springer, Heidelberg (2007)
124. Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.): RSFDGrC 2011. LNCS, vol. 6743. Springer, Heidelberg (2011)
125. Latkowski, R.: On decomposition for incomplete data. Fundamenta Informaticae 54(1), 1–16 (2003)
126. Latkowski, R.: Flexible indiscernibility relations for missing attribute values. Fundamenta Informaticae 67(1-3), 131–147 (2005)
127. Leibniz, G.W.: Discourse on metaphysics. In: Ariew and Garber [5], pp. 35–68
128. Leśniewski, S.: Grundzüge eines neuen Systems der Grundlagen der Mathematik. Fundamenta Mathematicae 14, 1–81 (1929)
129. Leśniewski, S.: On the foundations of mathematics. Topoi 2, 7–52 (1982)
130. Li, J., Cercone, N.: A rough set based model to rank the importance of association rules. In: Ślęzak et al. [321], pp. 109–118
131. Li, Y., Shiu, S.C.K., Pal, S.K., Liu, J.N.K.: A rough set-based case-based reasoner for text categorization. International Journal of Approximate Reasoning 41(2), 229–255 (2006)
132. Lin, T.Y.: Neighborhood systems and approximation in database and knowledge base systems. In: Emrich, M.L., Phifer, M.S., Hadzikadic, M., Raś, Z.W. (eds.) Proceedings of the Fourth International Symposium on Methodologies of Intelligent Systems (Poster Session), October 12-15, pp. 75–86. Oak Ridge National Laboratory, Charlotte (1989)
133. Lin, T.Y.: Special issue, Journal of the Intelligent Automation and Soft Computing 2(2) (1996)
134. Lin, T.Y.: The discovery, analysis and representation of data dependencies in databases. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1: Methodology and Applications. STUDFUZZ, pp. 107–121. Physica-Verlag, Heidelberg (1998)
135. Lin, T.Y., Cercone, N. (eds.): Rough Sets and Data Mining - Analysis of Imperfect Data. Kluwer Academic Publishers, Boston (1997)
136. Lin, T.Y., Wildberger, A.M. (eds.): Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery. Simulation Councils, Inc., San Diego (1995)

137. Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.): Rough Sets, Granular Computing and Data Mining. STUDFUZZ. Physica-Verlag, Heidelberg (2001)
138. Lingras, P.: Fuzzy - rough and rough - fuzzy serial combinations in neurocomputing. Neurocomputing 36(1-4), 29–44 (2001)
139. Lingras, P.: Unsupervised rough set classification using gas. Journal of Intelligent Information Systems 16(3), 215–228 (2001)
140. Lingras, P., West, C.: Interval set clustering of Web users with rough *K*-means. Journal of Intelligent Information Systems 23(1), 5–16 (2004)
141. Liu, J.: Autonomous Agents and Multi-Agent Systems: Explorations in Learning, Self-organization and Adaptive Computation. World Scientific Publishing (2001)
142. Łukasiewicz, J.: Die logischen Grundlagen der Wahrscheinlichkeitsrechnung, 1913. In: Borkowski, L. (ed.) Jan Łukasiewicz - Selected Works, pp. 16–63. North Holland Publishing Company, Polish Scientific Publishers, Amsterdam, Warsaw (1970)
143. Maimon, O., Rokach, L. (eds.): The Data Mining and Knowledge Discovery Handbook. Springer, Heidelberg (2005)
144. Maji, P., Pal, S.K.: Rough-Fuzzy Pattern Recognition: Application in Bioinformatics and Medical Imaging. Wiley Series in Bioinformatics. John Wiley & Sons, Hoboken (2012)
145. Marcus, S.: The paradox of the heap of grains, in respect to roughness, fuzziness and negligibility. In: Polkowski and Skowron [251], pp. 19–23
146. Marek, V.W., Rasiowa, H.: Approximating sets with equivalence relations. Theoretical Computer Science 48(3), 145–152 (1986)
147. Marek, V.W., Truszczyński, M.: Contributions to the theory of rough sets. Fundamenta Informaticae 39(4), 389–409 (1999)
148. McCarthy, J.: Notes on formalizing contex. In: Proceedings of the 13th International Joint Conference on Artifical Intelligence (IJCAI 1993), pp. 555–560. Morgan Kaufmann Publishers Inc., San Francisco (1993)
149. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: An experimental evaluation. Data Mining and Knowledge Discovery 14, 245–304 (2007)
150. Mill, J.S.: Ratiocinative and Inductive, Being a Connected View of the Principles of Evidence, and the Methods of Scientific Investigation. Parker, Son, and Bourn, West Strand London (1862)
151. Mitchel, T.M.: Machine Learning, Boston, MA. McGraw-Hill Series in Computer Science (1999)
152. Mitra, P., Mitra, S., Pal, S.K.: Modular rough fuzzy mlp: Evolutionary design. In: Skowron et al. [290], pp. 128–136
153. Mitra, P., Pal, S.K., Siddiqi, M.A.: Non-convex clustering using expectation maximization algorithm with rough set initialization. Pattern Recognition Letters 24(6), 863–873 (2003)
154. Moshkov, M., Skowron, A., Suraj, Z.: On testing membership to maximal consistent extensions of information systems. In: Greco et al. [72], pp. 85–90
155. Moshkov, M., Skowron, A., Suraj, Z.: On irreducible descriptive sets of attributes for information systems. In: Chan et al. [39], pp. 21–30
156. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: Partial Covers, Reducts and Decision Rules in Rough Sets - Theory and Applications. SCI, vol. 145. Springer, Heidelberg (2008)
157. Moshkov, M.J., Skowron, A., Suraj, Z.: On minimal rule sets for almost all binary information systems. Fundamenta Informaticae 80(1-3), 247–258 (2007)

158. Moshkov, M.J., Skowron, A., Suraj, Z.: On minimal inhibitory rules for almost all k-valued information systems. Fundamenta Informaticae 93(1-3), 261–272 (2009)
159. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning. SCI, vol. 360. Springer, Heidelberg (2011)
160. Nakamura, A.: Fuzzy quantifiers and rough quantifiers. In: Wang, P.P. (ed.) Advances in Fuzzy Theory and Technology II, pp. 111–131. Duke University Press, Durham (1994)
161. Nakamura, A.: On a logic of information for reasoning about knowledge. In: Ziarko [390], pp. 186–195
162. Nakamura, A.: A rough logic based on incomplete information and its application. International Journal of Approximate Reasoning 15(4), 367–378 (1996)
163. Nakata, M., Sakai, H.: Rough sets handling missing values probabilistically interpreted. In: Ślęzak et al. [320], pp. 325–334
164. Nguyen, H.S.: Discretization of real value attributes, boolean reasoning approach. Ph.D. Thesis, Warsaw University, Warsaw, Poland (1997)
165. Nguyen, H.S.: From optimal hyperplanes to optimal decision trees. Fundamenta Informaticae 34(1-2), 145–174 (1998)
166. Nguyen, H.S.: Efficient SQL-learning method for data mining in large data bases. In: Dean, T. (ed.) Sixteenth International Joint Conference on Artificial Intelligence IJCAI, pp. 806–811. Morgan-Kaufmann Publishers, Stockholm (1999)
167. Nguyen, H.S.: On efficient handling of continuous attributes in large data bases. Fundamenta Informaticae 48(1), 61–81 (2001)
168. Nguyen, H.S.: Approximate boolean reasoning approach to rough sets and data mining. In: Ślęzak et al. [321], pp. 12–22 (plenary talk)
169. Nguyen, H.S.: Approximate boolean reasoning: Foundations and applications in data mining. In: Peters and Skowron [228], pp. 344–523
170. Nguyen, H.S., Jankowski, A., Stepaniuk, J., Skowron, A., Szczuka, M.: Discovery of process models from data and domain knowledge: A rough-granular approach. In: Yao, J.T. (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, pp. 16–47. IGI Global, Hershey (2010)
171. Nguyen, H.S., Nguyen, S.H.: Pattern extraction from data. Fundamenta Informaticae 34, 129–144 (1998)
172. Nguyen, H.S., Nguyen, S.H.: Rough sets and association rule generation. Fundamenta Informaticae 40(4), 383–405 (1999)
173. Nguyen, H.S., Skowron, A.: Quantization of real value attributes. In: Proceedings of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, USA, pp. 34–37. Duke University, Durham (1995)
174. Nguyen, H.S., Skowron, A.: A rough granular computing in discovery of process models from data and domain knowledge. Journal of Chongqing University of Post and Telecommunications 20(3), 341–347 (2008)
175. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules - correspondence and complexity results. In: Skowron et al. [290], pp. 137–145
176. Nguyen, S.H.: Regularity analysis and its applications in data mining. In: Polkowski et al. [249], pp. 289–378
177. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered learning for concept synthesis. In: Peters and Skowron [233], pp. 187–208
178. Nguyen, S.H., Nguyen, H.S.: Some efficient algorithms for rough set methods. In: Sixth International Conference on Information Processing and Management of

Uncertainty on Knowledge Based Systems, IPMU 1996, Granada, Spain, vol. III, pp. 1451–1456 (1996)
179. Nguyen, T.T.: Eliciting domain knowledge in handwritten digit recognition. In: Pal et al. [193], pp. 762–767
180. Nguyen, T.T., Skowron, A.: Rough set approach to domain knowledge approximation. In: Wang et al. [367], pp. 221–228
181. Nguyen, T.T., Skowron, A.: Rough-granular computing in human-centric information processing. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing. SCI, vol. 182, pp. 1–30. Springer, Heidelberg (2009)
182. Noë, A.: Action in Perception. MIT Press (2004)
183. Omicini, A., Ricci, A., Viroli, M.: The multidisciplinary patterns of interaction from sciences to computer science. In: Goldin et al. [67], pp. 395–414
184. Orłowska, E.: Semantics of vague concepts. In: Dorn, G., Weingartner, P. (eds.) Foundation of Logic and Linguistics, pp. 465–482. Plenum Press, New York (1984)
185. Orłowska, E.: Rough concept logic. In: Skowron [277], pp. 177–186
186. Orłowska, E.: Reasoning about vague concepts. Bulletin of the Polish Academy of Sciences, Mathematics 35, 643–652 (1987)
187. Orłowska, E.: Logic for reasoning about knowledge. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 35, 559–572 (1989)
188. Orłowska, E.: Kripke semantics for knowledge representation logics. Studia Logica 49(2), 255–272 (1990)
189. Orłowska, E. (ed.): Incomplete Information: Rough Set Analysis. STUDFUZZ, vol. 13. Springer/Physica-Verlag, Heidelberg (1997)
190. Orłowska, E., Pawlak, Z.: Representation of non-deterministic information. Theoretical Computer Science 29, 27–39 (1984)
191. Orłowska, E., Peters, J.F., Rozenberg, G., Skowron, A.: Special volume dedicated to the memory of Zdzisław Pawlak, Fundamenta Informaticae 75(1-4) (2007)
192. Pal, S.: Computational theory perception (ctp), rough-fuzzy uncertainty analysis and mining in bioinformatics and web intelligence: A unified framework. In: Peters and Skowron [230], pp. 106–129
193. Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.): PReMI 2005. LNCS, vol. 3776. Springer, Heidelberg (2005)
194. Pal, S.K., Dasgupta, B., Mitra, P.: Rough self organizing map. Applied Intelligence 21, 289–299 (2004)
195. Pal, S.K., Mitra, P.: Case generation using rough sets with fuzzy representation. IEEE Trans. Knowledge and Data Engineering 16(3), 292–300 (2004)
196. Pal, S.K., Mitra, P.: Pattern Recognition Algorithms for Data Mining. CRC Press, Boca Raton (2004)
197. Pal, S.K., Pedrycz, W., Skowron, A., Swiniarski, R.: Special volume: Rough-neuro computing, Neurocomputing 36 (2001)
198. Pal, S.K., Peters, J.F. (eds.): Rough Fuzzy Image Analysis Foundations and Methodologies. Chapman & Hall/CRC, Boca Raton, Fl (2010)
199. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2004)
200. Pal, S.K., Skowron, A. (eds.): Rough Fuzzy Hybridization: A New Trend in Decision-Making. Springer, Singapore (1999)
201. Pancerz, K., Suraj, Z.: Modelling concurrent systems specified by dynamic information systems: A rough set approach. Electronic Notes in Theoretical Computer Science 82(4), 206–218 (2003)

202. Pancerz, K., Suraj, Z.: Discovering concurrent models from data tables with the ROSECON system. Fundamenta Informaticae 60(1-4), 251–268 (2004)
203. Pancerz, K., Suraj, Z.: Discovering concurrent models from data tables with the ROSECON system. Fundamenta Informaticae 60(1-4), 251–268 (2004)
204. Pancerz, K., Suraj, Z.: Discovery of asynchronous concurrent models from experimental tables. Fundamenta Informaticae 61(2), 97–116 (2004)
205. Pancerz, K., Suraj, Z.: Restriction-based concurrent system design using the rough set formalism. Fundamenta Informaticae 67(1-3), 233–247 (2005)
206. Pancerz, K., Suraj, Z.: Reconstruction of concurrent system models described by decomposed data tables. Fundamenta Informaticae 71(1), 121–137 (2006)
207. Pancerz, K., Suraj, Z.: Towards efficient computing consistent and partially consistent extensions of information systems. Fundamenta Informaticae 79(3-4), 553–566 (2007)
208. Papageorgiou, E.I., Stylios, C.D.: Fuzzy cognitive maps. In: Pedrycz et al. [223], pp. 755–774
209. Pawlak, Z.: Classification of Objects by Means of Attributes, Reports. Institute of Computer Science, vol. 429. Polish Academy of Sciences, Warsaw (1981)
210. Pawlak, Z.: Information systems - theoretical foundations. Information Systems 6, 205–218 (1981)
211. Pawlak, Z.: Rough Relations, Reports. Institute of Computer Science, vol. 435. Polish Academy of Sciences, Warsaw (1981)
212. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
213. Pawlak, Z.: Rough logic. Bulletin of the Polish Academy of Sciences, Technical Sciences 35(5-6), 253–258 (1987)
214. Pawlak, Z.: Decision logic. Bulletin of the EATCS 44, 201–225 (1991)
215. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. In: System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
216. Pawlak, Z.: Concurrent versus sequential - the rough sets perspective. Bulletin of the EATCS 48, 178–190 (1992)
217. Pawlak, Z.: Decision rules, Bayes' rule and rough sets. In: Skowron et al. [290], pp. 1–9
218. Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets and rough logic: A KDD perspective. In: Polkowski et al. [249], pp. 583–646
219. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 251–271. John Wiley & Sons, New York (1994)
220. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177(1), 41–73 (2007)
221. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177(28-40), 1 (2007)
222. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
223. Pedrycz, W., Skowron, S., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, Hoboken (2008)
224. Peters, G., Lingras, P., Ślęzak, D., Yao, Y.Y. (eds.): Rough Sets: Selected Methods and Applications in Management and Engineering. Advanced Information and Knowledge Processing. Springer, Heidelberg (2012)
225. Peters, J., Skowron, A.: Special issue on a rough set approach to reasoning about data, International Journal of Intelligent Systems 16(1) (2001)

226. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets III. LNCS, vol. 3400. Springer, Heidelberg (2005)
227. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets IV. LNCS, vol. 3700. Springer, Heidelberg (2005)
228. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets V. LNCS, vol. 4100. Springer, Heidelberg (2006)
229. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets VIII. LNCS, vol. 5084. Springer, Heidelberg (2008)
230. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets XI. LNCS, vol. 5946. Springer, Heidelberg (2010)
231. Peters, J.F., Skowron, A., Dubois, D., Grzymała-Busse, J.W., Inuiguchi, M., Polkowski, L. (eds.): Transactions on Rough Sets II. LNCS, vol. 3135. Springer, Heidelberg (2004)
232. Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.): Transactions on Rough Sets VI. LNCS, vol. 4374. Springer, Heidelberg (2007)
233. Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.): Transactions on Rough Sets I. LNCS, vol. 3100. Springer, Heidelberg (2004)
234. Peters, J.F., Skowron, A., Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.): Transactions on Rough Sets XIII. LNCS, vol. 6499. Springer, Heidelberg (2011)
235. Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.): Transactions on Rough Sets VII. LNCS, vol. 4400. Springer, Heidelberg (2007)
236. Peters, J.F., Skowron, A., Rybiński, H. (eds.): Transactions on Rough Sets IX. LNCS, vol. 5390. Springer, Heidelberg (2008)
237. Peters, J.F., Skowron, A., Sakai, H., Chakraborty, M.K., Slęzak, D., Hassanien, A.E., Zhu, W. (eds.): Transactions on Rough Sets XIV. LNCS, vol. 6600. Springer, Heidelberg (2011)
238. Peters, J.F., Skowron, A., Słowiński, R., Lingras, P., Miao, D., Tsumoto, S. (eds.): Transactions on Rough Sets XII. LNCS, vol. 6190. Springer, Heidelberg (2010)
239. Peters, J.F., Skowron, A., Suraj, Z.: An application of rough set methods in control design. Fundamenta Informaticae 43(1-4), 269–290 (2000)
240. Peters, J.F., Skowron, A., Suraj, Z.: An application of rough set methods in control design. Fundamenta Informaticae 43(1-4), 269–290 (2000)
241. Peters, J.F., Skowron, A., Wolski, M., Chakraborty, M.K., Wu, W.-Z. (eds.): Transactions on Rough Sets X. LNCS, vol. 5656. Springer, Heidelberg (2009)
242. Peters, J.F., Suraj, Z., Shan, S., Ramanna, S., Pedrycz, W., Pizzi, N.J.: Classification of meteorological volumetric radar data using rough set methods. Pattern Recognition Letters 24(6), 911–920 (2003)
243. Pindur, R., Susmaga, R., Stefanowski, J.: Hyperplane aggregation of dominance decision rules. Fundamenta Informaticae 61(2), 117–137 (2004)
244. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the AMS 50(5), 537–544 (2003)
245. Polkowski, L.: Rough Sets: Mathematical Foundations. Advances in Soft Computing. Physica-Verlag, Heidelberg (2002)
246. Polkowski, L.: Rough mereology: A rough set paradigm for unifying rough set theory and fuzzy set theory. Fundamenta Informaticae 54, 67–88 (2003)
247. Polkowski, L.: A note on 3-valued rough logic accepting decision rules. Fundamenta Informaticae 61(1), 37–45 (2004)

248. Polkowski, L.: Approximate Reasoning by Parts. An Introduction to Rough Mereology. ISRL, vol. 20. Springer, Heidelberg (2011)
249. Polkowski, L., Lin, T.Y., Tsumoto, S. (eds.): Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. STUDFUZZ, vol. 56. Springer/Physica-Verlag, Heidelberg (2000)
250. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning 15(4), 333–365 (1996)
251. Polkowski, L., Skowron, A. (eds.): RSCTC 1998. LNCS (LNAI), vol. 1424. Springer, Heidelberg (1998)
252. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery 1: Methodology and Applications. STUDFUZZ, vol. 18. Physica-Verlag, Heidelberg (1998)
253. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. STUDFUZZ, vol. 19. Physica-Verlag, Heidelberg (1998)
254. Polkowski, L., Skowron, A.: Towards adaptive calculus of granules. In: Zadeh, L.A., Kacprzyk, J. (eds.) Computing with Words in Information/Intelligent Systems, pp. 201–227. Physica-Verlag, Heidelberg (1999)
255. Polkowski, L., Skowron, A.: Rough mereological calculi of granules: A rough set approach to computation. Computational Intelligence: An International Journal 17(3), 472–492 (2001)
256. Polkowski, L., Skowron, A., Żytkow, J.: Rough foundations for rough sets. In: Lin and Wildberger [136], pp. 55–58
257. Quafafou, M., Boussouf, M.: Generalized rough sets based feature selection. Intelligent Data Analysis 4(1), 3–17 (2000)
258. Ramsay, J.O., Silverman, B.W.: Applied Functional Data Analysis. Springer, Berlin (2002)
259. Rasiowa, H.: Axiomatization and completeness of uncountably valued approximation logic. Studia Logica 53(1), 137–160 (1994)
260. Rasiowa, H., Skowron, A.: Approximation logic. In: Bibel, W., Jantke, K.P. (eds.) Mathematical Methods of Specification and Synthesis of Software Systems. Mathematical Research, vol. 31, pp. 123–139. Akademie Verlag, Berlin (1985)
261. Rasiowa, H., Skowron, A.: Rough concept logic. In: Skowron [277], pp. 288–297
262. Rauszer, C.: An equivalence between indiscernibility relations in information systems and a fragment of intuitionistic logic. In: Skowron [277], pp. 298–317
263. Rauszer, C.: An equivalence between theory of functional dependence and a fragment of intuitionistic logic. Bulletin of the Polish Academy of Sciences, Mathematics 33, 571–579 (1985)
264. Rauszer, C.: Logic for information systems. Fundamenta Informaticae 16, 371–382 (1992)
265. Rauszer, C.: Knowledge representation systems for groups of agents. In: Wroński, J. (ed.) Philosophical Logic in Poland, pp. 217–238. Kluwer, Dordrecht (1994)
266. Read, S.: Thinking about Logic: An Introduction to the Philosophy of Logic. Oxford University Press, Oxford (1994)
267. Rissanen, J.: Modelling by shortes data description. Automatica 14, 465–471 (1978)
268. Rissanen, J.: Minimum-description-length principle. In: Kotz, S., Johnson, N. (eds.) Encyclopedia of Statistical Sciences, pp. 523–527. John Wiley & Sons, New York (1985)
269. Roddick, J., Hornsby, K.S., Spiliopoulou, M.: An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. In: Roddick, J., Hornsby, K.S. (eds.) TSDM 2000. LNCS (LNAI), vol. 2007, pp. 147–164. Springer, Heidelberg (2001)

270. Roy, A., Pal, S.K.: Fuzzy discretization of feature space for a rough set classifier. Pattern Recognition Letters 24(6), 895–902 (2003)
271. Russell, B.: An Inquiry into Meaning and Truth. George Allen & Unwin Ltd. and W. W. Norton, London and New York (1940)
272. Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.): RSFDGrC 2009. LNCS, vol. 5908. Springer, Heidelberg (2009)
273. Serafini, L., Bouquet, P.: Comparing formal theories of context in ai. Artificial Intelligence 155, 41–67 (2004)
274. Sever, H., Raghavan, V.V., Johnsten, T.D.: The status of research on rough sets for knowledge discovery in databases. In: Sivasundaram, S. (ed.) Proceedings of the Second Internationall Conference On Nonlinear Problems in Aviation and Aerospace (ICNPAA 1998), Daytona Beach, FL, April 29-May 1, vol. 2, pp. 673–680. Embry-Riddle Aeronautical University, Daytona Beach (1998)
275. Shan, N., Ziarko, W.: An incremental learning algorithm for constructing decision rules. In: Ziarko, W. (ed.) Rough Sets, Fuzzy Sets and Knowledge Discovery, pp. 326–334. Springer, Berlin (1994)
276. Skowron, A.: Rough Sets in Perception-Based Computing (Keynote Talk). In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 21–29. Springer, Heidelberg (2005)
277. Skowron, A. (ed.): SCT 1984. LNCS, vol. 208. Springer, Heidelberg (1985)
278. Skowron, A.: Boolean Reasoning for Decision Rules Generation. In: Komorowski, J., Raś, Z.W. (eds.) ISMIS 1993. LNCS, vol. 689, pp. 295–305. Springer, Heidelberg (1993)
279. Skowron, A.: Extracting laws from decision tables. Computational Intelligence: An International Journal 11, 371–388 (1995)
280. Skowron, A.: Synthesis of adaptive decision systems from experimental data. In: Aamodt, A., Komorowski, J. (eds.) Fifth Scandinavian Conference on Artificial Intelligence SCAI 1995. Frontiers in Artificial Intelligence and Applications, vol. 28, pp. 220–238. IOS Press, Trondheim (1995)
281. Skowron, A.: Rough sets in KDD - plenary talk. In: Shi, Z., Faltings, B., Musen, M. (eds.) 16-th World Computer Congress (IFIP 2000): Proceedings of Conference on Intelligent Information Processing (IIP 2000), pp. 1–14. Publishing House of Electronic Industry, Beijing (2000)
282. Skowron, A.: Approximate reasoning by agents in distributed environments. In: Zhong, N., Liu, J., Ohsuga, S., Bradshaw, J. (eds.) Intelligent Agent Technology Research and Development: Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology IAT 2001, Maebashi, Japan, October 23-26, pp. 28–39. World Scientific, Singapore (2001)
283. Skowron, A.: Rough sets and boolean reasoning. In: Pedrycz, W. (ed.) Granular Computing: an Emerging Paradigm. STUDFUZZ, vol. 70, pp. 95–124. Springer/Physica-Verlag, Heidelberg (2001)
284. Skowron, A.: Toward intelligent systems: Calculi of information granules. Bulletin of the International Rough Set Society 5(1-2), 9–30 (2001)
285. Skowron, A.: Approximate reasoning in distributed environments. In: Zhong and Liu [387], pp. 433–474
286. Skowron, A.: Perception logic in intelligent systems (keynote talk). In: Blair, S., et al. (eds.) Proceedings of the 8th Joint Conference on Information Sciences (JCIS 2005), Salt Lake City, Utah, USA, July 21-26, vol. 8, pp. 1–5. X-CD Technologies: A Conference & Management Company, 15 Coldwater Road (2005)

287. Skowron, A.: Rough sets and vague concepts. Fundamenta Informaticae 64(1-4), 417–431 (2005)
288. Skowron, A., Grzymała-Busse, J.W.: From rough set theory to evidence theory. In: Yager, R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 193–236. John Wiley & Sons, New York (1994)
289. Skowron, A., Nguyen, H.S.: Boolean Reasoning Scheme with Some Applications in Data Mining. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 107–115. Springer, Heidelberg (1999)
290. Zhong, N., Skowron, A., Ohsuga, S. (eds.): RSFDGrC 1999. LNCS (LNAI), vol. 1711. Springer, Heidelberg (1999)
291. Skowron, A., Pal, S.K.: Special volume: Rough sets, pattern recognition and data mining, Pattern Recognition Letters 24(6) (2003)
292. Skowron, A., Pal, S.K., Nguyen, H.S.: Special issue: Rough sets and fuzzy sets in natural computing, Theoretical Computer Science 412(42) (2011)
293. Skowron, A., Pawlak, Z., Komorowski, J., Polkowski, L.: A rough set perspective on data and knowledge. In: Kloesgen, W., Żytkow, J. (eds.) Handbook of KDD, pp. 134–149. Oxford University Press, Oxford (2002)
294. Skowron, A., Peters, J.: Rough sets: Trends and challenges. In: Wang et al. [367], pp. 25–34 (plenary talk)
295. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński [324], pp. 331–362
296. Skowron, A., Stepaniuk, J.: Generalized approximation spaces. In: The Third International Workshop on Rough Sets and Soft Computing Proceedings (RSSC 1994), San Jose, California, USA, November 10-12, pp. 156–163. San Jose University, San Jose (1994)
297. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
298. Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular computing. International Journal of Intelligent Systems 16(1), 57–86 (2001)
299. Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: Pal et al. [199], pp. 43–84
300. Skowron, A., Stepaniuk, J.: Ontological framework for approximation. In: Ślęzak et al. [320], pp. 718–727
301. Skowron, A., Stepaniuk, J.: Approximation spaces in rough-granular computing. Fundamenta Informaticae 100, 141–157 (2010)
302. Skowron, A., Stepaniuk, J., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72, 363–378 (2006)
303. Skowron, A., Stepaniuk, J., Swiniarski, R.: Modelling rough granular computing based on approximation spaces. Information Sciences 184, 20–43 (2012)
304. Skowron, A., Suraj, Z.: A rough set approach to real-time state identification. Bulletin of the EATCS 50, 264–275 (1993)
305. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences, Technical Sciences 41, 237–254 (1993)
306. Skowron, A., Suraj, Z.: Discovery of concurrent data models from experimental tables: A rough set approach. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD 1995), Montreal, Canada, August 20-21, pp. 288–293. AAAI Press, Menlo Park (1995)
307. Skowron, A., Swiniarski, R.: Rough sets and higher order vagueness. In: Ślęzak et al. [320], pp. 33–42

308. Skowron, A., Swiniarski, R., Synak, P.: Approximation spaces and information granulation. In: Peters and Skowron [226], pp. 175–189
309. Skowron, A., Synak, P.: Complex patterns. Fundamenta Informaticae 60(1-4), 351–366 (2004)
310. Skowron, A., Szczuka, M. (eds.): Proceedings of the Workshop on Rough Sets in Knowledge Discovery and Soft Computing at ETAPS 2003. Electronic Notes in Computer Science, vol. 82(4). Elsevier, Amsterdam (2003), www.elsevier.nl/locate/entcs/volume82.html
311. Skowron, A., Szczuka, M.: Toward Interactive Computations: A Rough-Granular Approach. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning II. SCI, vol. 263, pp. 23–42. Springer, Heidelberg (2010)
312. Skowron, A., Wasilewski, P.: Information systems in modelling interactive computations on granules. Theoretical Computer Science 412(42), 5939–5959 (2011)
313. Skowron, A., Wasilewski, P.: Toward interactive rough-granular computing. Control & Cybernetics 40(2), 1–23 (2011)
314. Ślęzak, D.: Approximate reducts in decision tables. In: Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU 1996, Granada, Spain, vol. III, pp. 1159–1164 (1996)
315. Ślęzak, D.: Association Reducts: A Framework for Mining Multi-attribute Dependencies. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 354–363. Springer, Heidelberg (2005)
316. Ślęzak, D.: Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44, 291–319 (2000)
317. Ślęzak, D.: Various approaches to reasoning with frequency-based decision reducts: A survey. In: Polkowski et al. [249], pp. 235–285
318. Ślęzak, D.: Approximate entropy reducts. undamenta Informaticae 53, 365–387 (2002)
319. Ślęzak, D.: Rough sets and Bayes factor. In: Peters and Skowron [226], pp. 202–229
320. Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.): RSFDGrC 2005, Part I. LNCS (LNAI), vol. 3641. Springer, Heidelberg (2005)
321. Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.): RSFDGrC 2005, Part II. LNCS (LNAI), vol. 3642. Springer, Heidelberg (2005)
322. Ślęzak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
323. Słowiński, R.: New Applications and Theoretical Foundations of the Dominance-based Rough Set Approach. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 2–3. Springer, Heidelberg (2010)
324. Słowiński, R. (ed.): Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory. System Theory, Knowledge Engineering and Problem Solving, vol. 11. Kluwer Academic Publishers, Dordrecht (1992)
325. Słowiński, R., Greco, S., Matarazzo, B.: Rough set analysis of preference-ordered data. In: Alpigini et al. [2], pp. 44–59
326. Słowiński, R., Stefanowski, J. (eds.): Special issue: Proceedings of the First International Workshop on Rough Sets: State of the Art and Perspectives, Kiekrz, Poznań, Poland, September 2-4, 1992. Foundations of Computing and Decision Sciences, vol. 18(3-4) (1993)
327. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co. (2000)
328. Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems. Springer, Heidelberg (2004)

329. Stepaniuk, J.: Approximation spaces, reducts and representatives. In: Polkowski and Skowron [253], pp. 109–126
330. Stepaniuk, J.: Knowledge discovery by application of rough set models. In: Polkowski et al. [249], pp. 137–233
331. Stepaniuk, J. (ed.): Rough-Granular Computing in Knowledge Discovery and Data Mining. Springer, Heidelberg (2008)
332. Stone, P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. The MIT Press, Cambridge (2000)
333. Strąkowski, T., Rybiński, H.: A new approach to distributed algorithms for reduct calculation. In: Peters and Skowron [236], pp. 365–378
334. Suraj, Z.: Discovery of concurrent data models from experimental tables: A rough set approach. Fundamenta Informaticae 28(3-4), 353–376 (1996)
335. Suraj, Z.: Rough set methods for the synthesis and analysis of concurrent processes. In: Polkowski et al. [249], pp. 379–488
336. Suraj, Z.: Discovering concurrent process models in data: A rough set approach. In: Sakai et al. [272], pp. 12–19
337. Suraj, Z., Pancerz, K.: A synthesis of concurrent systems: A rough set approach. In: Wang et al. [367], pp. 299–302
338. Suraj, Z., Pancerz, K.: The ROSECON system - a computer tool for modelling and analysing of processes. In: 2005 International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA 2005), International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC 2005), Vienna, Austria, November 28-30, pp. 829–834. IEEE Computer Society (2005)
339. Suraj, Z., Pancerz, K.: Some remarks on computing consistent extensions of dynamic information systems. In: Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications (ISDA 2005), Wrocław, Poland, September 8-10, pp. 420–425. IEEE Computer Society (2005)
340. Suraj, Z., Pancerz, K., Owsiany, G.: On consistent and partially consistent extensions of information systems. In: Ślęzak et al. [320], pp. 224–233
341. Swift, J.: Gulliver's Travels into Several Remote Nations of the World. Ananymous Publisher, London (1726)
342. Swiniarski, R.: Rough sets and principal component analysis and their applications. data model building and classification. In: Pal and Skowron [200], pp. 275–300
343. Swiniarski, R.: An application of rough sets and Haar wavelets to face recognition. In: Ziarko and Yao [394], pp. 561–568
344. Swiniarski, R., Hargis, L.: A new halftoning method based on error diffusion with rough set filterin. In: Polkowski and Skowron [253], pp. 336–342
345. Swiniarski, R., Skowron, A.: Rough set methods in feature selection and extraction. Pattern Recognition Letters 24(6), 833–849 (2003)
346. Swiniarski, R.W., Pancerz, K., Suraj, Z.: Prediction of model changes of concurrent systems described by temporal information systems. In: Proceedings of The 2005 International Conference on Data Mining (DMIN 2005), Las Vegas, Nevada, USA, June 20-23, pp. 51–57. CSREA Press (2005)
347. Swiniarski, R.W., Skowron, A.: Independent component analysis, principal component analysis and rough sets in face recognition. In: Peters and Skowron [233], pp. 392–404
348. Sycara, K.: Multiagent systems. AI Magazine pp. 79–92 (Summer 1998)

349. Szczuka, M., Skowron, A., Stepaniuk, J.: Function approximation and quality measures in rough-granular systems. Fundamenta Informaticae 109(3-4), 339–354 (2011)
350. Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.): RSCTC 2010. LNCS (LNAI), vol. 6086. Springer, Heidelberg (2010)
351. Tarski, A.: Logic, Semantics, Metamathematics. Oxford University Press, Oxford (1983) (translated by J. H. Woodger)
352. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional Learning of Spatio-temporal Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010)
353. Terano, T., Nishida, T., Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T. (eds.): JSAI-WS 2001. LNCS (LNAI), vol. 2253. Springer, Heidelberg (2001)
354. Torra, V., Narukawa, Y.: Modelling Decisions Information Fusion and Aggregation Operators. Springer, Heidelberg (2007)
355. Tsumoto, S., Kobayashi, S., Yokomori, T., Tanaka, H., Nakamura, A. (eds.): Proceedings of the The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, University of Tokyo, Japan, November 6-8. The University of Tokyo, Tokyo (1996)
356. Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.): RSCTC 2004. LNCS (LNAI), vol. 3066. Springer, Heidelberg (2004)
357. Tsumoto, S., Tanaka, H.: PRIMEROSE: Probabilistic rule induction method based on rough sets and resampling methods. Computational Intelligence: An International Journal 11, 389–405 (1995)
358. Unnikrishnan, K.P., Ramakrishnan, N., Sastry, P.S., Uthurusamy, R.: Network reconstruction from dynamic data. SIGKDD Explorations 8(2), 90–91 (2006)
359. Vakarelov, D.: A modal logic for similarity relations in Pawlak knowledge representation systems. Fundamenta Informaticae 15(1), 61–79 (1991)
360. Vakarelov, D.: Modal logics for knowledge representation systems. Theoretical Computer Science 90(2), 433–456 (1991)
361. Vakarelov, D.: A duality between Pawlak's knowledge representation systems and bi-consequence systems. Studia Logica 55(1), 205–228 (1995)
362. Vakarelov, D.: A modal characterization of indiscernibility and similarity relations in Pawlak's information systems. In: Ślęzak et al. [320], pp. 12–22 (plenary talk)
363. van der Aalst, W.M.P. (ed.): Process Mining Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011)
364. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
365. Vitória, A.: A framework for reasoning with rough sets. Licentiate Thesis, Linköping University 2004. In: Peters and Skowron [227], pp. 178–276
366. Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.): RSKT 2008. LNCS (LNAI), vol. 5009. Springer, Heidelberg (2008)
367. Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.): RSFDGrC 2003. LNCS (LNAI), vol. 2639. Springer, Heidelberg (2003)
368. Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.): RSKT 2006. LNCS (LNAI), vol. 4062. Springer, Heidelberg (2006)
369. Wang, J., Jia, C., Zhao, K.: Investigation on AQ11, ID3 and the principle of discernibility matrix. Journal of Computer Science and Technology 16(1), 1–12 (2001)
370. Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.): RSKT 2009. LNCS, vol. 5589. Springer, Heidelberg (2009)
371. Wojna, A.: Analogy based reasoning in classifier construction. In: Peters and Skowron [227], pp. 277–374

372. Wong, S.K.M., Ziarko, W.: Comparison of the probabilistic approximate classification and the fuzzy model. Fuzzy Sets and Systems 21, 357–362 (1987)
373. Wróblewski, J.: Theoretical foundations of order-based genetic algorithms. Fundamenta Informaticae 28, 423–430 (1996)
374. Wróblewski, J.: Analyzing relational databases using rough set based methods. In: Eighth International Conference on Processing and Management of Uncertainty in Knowledge-Based Systems IPMU, Madrid, Spain, vol. I, pp. 256–262 (2000)
375. Wróblewski, J.: Adaptive aspects of combining approximation spaces. In: Pal et al. [199], pp. 139–156
376. Wu, F.X.: Inference of gene regulatory networks and its validation. Current Bioinformatics 2(2), 139–144 (2007)
377. Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D.: RSKT 2007. LNCS (LNAI), vol. 4481. Springer, Heidelberg (2007)
378. Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.): RSKT 2011. LNCS (LNAI), vol. 6954. Springer, Heidelberg (2011)
379. Yao, Y.Y.: Generalized rough set models. In: Polkowski and Skowron [252], pp. 286–318
380. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16, 87–104 (2001)
381. Yao, Y.Y.: On generalizing rough set theory. In: Wang et al. [367], pp. 44–51
382. Yao, Y.Y.: Probabilistic approaches to rough sets. Expert Systems 20, 287–297 (2003)
383. Yao, Y.Y., Wong, S.K.M., Lin, T.Y.: A review of rough set models. In: Lin and Cercone [135], pp. 47–75
384. Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.): RSKT 2010. LNCS, vol. 6401. Springer, Heidelberg (2010)
385. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
386. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)
387. Zhong, N., Liu, J. (eds.): Intelligent Technologies for Information Analysis. Springer, Heidelberg (2004)
388. Zhu, W.: Topological approaches to covering rough sets. Information Sciences 177, 1499–1508 (2007)
389. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)
390. Ziarko, W. (ed.): Rough Sets, Fuzzy Sets and Knowledge Discovery: Proceedings of the Second International Workshop on Rough Sets and Knowledge Discovery (RSKD 1993), Banff, Alberta, Canada, October 12-15. Workshops in Computing. Springer & British Computer Society, London, Berlin (1993)
391. Ziarko, W.: Special issue, Computational Intelligence: An International Journal 11(2) (1995)
392. Ziarko, W.: Special issue, Fundamenta Informaticae 27(2-3) (1996)
393. Ziarko, W.: Probabilistic decision tables in the variable precision rough set model. Computational Intelligence 17, 593–603 (2001)
394. Ziarko, W.P., Yao, Y. (eds.): RSCTC 2000. LNCS (LNAI), vol. 2005. Springer, Heidelberg (2001)

Chapter 4
Zdzisław Pawlak, Databases and Rough Sets

Victor W. Marek

Abstract. We discuss work of Zdzisław Pawlak in the area of databases and the extension of that work to the theory of rough sets. In particular, we look at his motivations for introducing information storage and retrieval systems and how this, eventually, led to rough sets theory.

Keywords: Information storage and retrieval systems (i.s.r.), rough sets.

4.1 Introduction

In this memoir, I am recalling my collaboration with Professor Zdzisław Pawlak, especially during 1970s and early 1980s. This period coincides with two ideas that originated with Pawlak during that time: a model of databases (it was called *information storage and retrieval systems* and was pursued by a group of scientists in Poland and in other research centers mostly in Eastern Europe) and then later work on approximating sets (of records or other objects) by means of some pairs of sets of objects. This latter theory is now called *rough sets* and again originated with Pawlak. The reason why I write about these areas is that during that specific period I was a close collaborator of Pawlak and worked with him on a variety of projects related to these two areas.

Of course, Pawlak studied many other areas of Computer Science and, more generally, Mathematics. Specifically he contributed to the area of models of computation, data structures, combinatorial optimization, and theory of conflicts – to name a few. I am sure these contributions will be discussed by others, and I will focus on information storage and retrieval systems and on rough sets.

Victor W. Marek
Department of Computer Science
University of Kentucky, Lexington, KY 40506-0633, USA
e-mail: marek@cs.uky.edu

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 175–184.
springerlink.com

Let me first describe how this all started. In 1960, fresh from high school, I started studies of Mathematics at Warsaw University. There was no Computer Science program at Warsaw at the time, but some aspects of Computer Science were taught in the numerical analysis program that was part of Mathematics. Very soon, I was attracted to Foundations of Mathematics. Warsaw, of course, had a strong tradition of Foundations. There were several groups of researchers pursuing foundational studies. The strongest group was centered around Professor Andrzej Mostowski who was both the head of Foundations section at the Mathematical Institute of Polish Academy of Sciences and, at the same time, the Chair of Algebra at the University. Other notable logicians at the time included Professor Helena Rasiowa (Chair of Logic at the Warsaw University and later close collaborator of Pawlak), Professor Andrzej Grzegorczyk (first at the University, then at Polish Academy of Science), Professor Wanda Szmielew (Chair of Foundations of Geometry at the University), and Professor Jerzy Łoś. Professors Rasiowa and Grzegorczyk were Mostowski's students. Professor Szmielew was Alfred Tarski's student. Although Tarski (since the middle of WW II) was at the University of California, Berkeley, he somehow influenced Warsaw foundational research – in spite of the fact that at the time a "Cold War" was raging between the countries dependent of Soviet Union and so-called the West and so communications were sporadic, censored, and slow. In addition to the forenamed researchers, I soon met two other: One was Andrzej Ehrenfeucht and the other Zdzisław Pawlak. Both, at the time, were working at the Mathematical Institute of Polish Academy of Sciences. Zdzisław was a computer engineer, and Andrzej was a logician, with clear interests in Foundations.

There were several opportunities for participation in classes and seminars devoted to Foundations. Each of the principals mentioned above taught some lectures and lead seminar series. As a sophomore and then a junior at the University I joined two. One was the General Foundations Seminar, usually convening on Wednesday, 5 pm (essentially in Tarskian tradition). That seminar was lead by a distant (at least at that time) figure – Professor Andrzej Mostowski. All current major results in Foundations were presented there. There were also other series. Besides of Professor Rasiowa's seminar (dominated by the algebraic approach to logic), Professor Szmielew's seminar (Foundations of Geometry) and, occasionally, Professor Grzegorczyk's seminar, there was Ehrenfeucht and Pawlak seminar at the Mathematical Institute. The atmosphere there was very informal. Unlike in other seminars the "thou" form was used there, and this informality and relaxed atmosphere certainly appealed to people like myself. The audience was, unlike in other seminars, very diverse: logicians, probabilists, philosophers, computer scientists and even medical researchers. A number of papers devoted to automated theorem proving were read there. I remember two – pioneering Hao Wang work on proving tautologies using computers and Davis and Putnam work on resolution. The memories of events that happened some 50 years ago are blurred; somehow, the presentation of Professor Ewa Orłowska comes to mind.

Very soon, I started to talk regularly to Andrzej Ehrenfeucht and, eventually, at his suggestion and with his guidance I wrote a master's (M.Sc) thesis. Like Mostowski (but unlike the others), Andrzej was a "generalist" of Foundations - he had extensive

knowledge of all major areas of Foundations: Proof Theory, Set Theory, Recursion Theory, and especially Model Theory. For reasons not entirely clear today, Andrzej suggested for my M.Sc. thesis a topic from Combinatorial Set Theory. Maybe the reason was that he heard a talk on set-theoretical topology and immediately saw a generalization? Maybe he heard it from someone who heard it from the great combinatorist Paul Erdös? Anyway, as I recall, we were sitting in a cafe at Marszałkowska Street in Warsaw (close to Constitution Square) eating cakes, drinking coffee (likely Andrzej also drank cognac - he could afford it and he liked it then) and Andrzej suggested a problem closely related to so-called Δ-lemma. He told me to use a specific form of induction. I was reading two-year Mostowski's course in Set Theory and it fit together very well. Soon the problem was solved, publication written, thesis defended, and as a result I became a teaching assistant at Mostowski's group. The year was 1964, and at exactly that year a breakthrough in Foundations of Set Theory occurred - Paul J. Cohen of Stanford University invented a new technique called "forcing". He utilized it to prove independence of Continuum Hypothesis, a problem stemming out of famous Hilbert problems. We, in the vicinity of Mostowski, dropped everything and started to research the area of Foundations of Set Theory. So, naturally, Foundations of Computer Science went (fortunately temporarily for me) away. Moreover, soon Andrzej Ehrenfeucht left for United States and after short stint in California settled in Colorado. He is not a person who reads or writes much. Actually, to this day, he writes almost nothing, but magically knows much. This lack of communication created a vacuum, at least for me and for a couple of years I did exotic things like studies of second-order arithmetic (what is it?), constructible hierarchy (even worse!), and other fashionable, but remote from Computer Science areas. After getting a PhD degree in 1968 and post-doc'ing in Holland in 1970/71, as I was returning to Poland I took a detour and visited Janusz Onyszkiewicz (another Warsaw logician) who was at Aarhus University in Denmark for a year. I noticed there that logicians were deeply engaged in various problems stemming from Computer Science. This must have influenced me somehow because once back in Warsaw, in addition to the research discussed above, I started to look at the areas further away from Mostowski-style Foundations. Soon a series of phone conversations with Zdzisław on some issues related to something related to databases followed. I knew very little about databases, and at the time I was not familiar with the work of E.F. Codd on relational model of databases. Worse, I did not understand the issues. Likely, nobody in Warsaw did. That is, except Zdzisław. He somehow knew that the time to apply various techniques of logic to databases has come. In the next section I will describe the work done by Zdzisław and myself, and how it is related to databases as we know them now. This research coincided with the significant changes in the attitude of Polish scientific community toward Computer Science. In rapid succession, and obviously Zdzisław was instrumental in these things happening, several things occurred. First, Computer Science program was established at the Warsaw University (Professor Rasiowa, the then Dean of Mathematics and Physics, later Mathematics and Mechanics was also deeply involved). Second, the Academy of Sciences converted its "Computational Center" into a Computer Science Institute (the formal change, including the name change, came later), thus creating another

place where Computer Science researchers could be employed. There was yet another important change. Warsaw Technical University opened a highly competitive program called "Technical Physics and Applied Mathematics" which was, in reality, Computer Science. That program, due to its competitiveness and prestige, attracted a cream of computationally minded young students from all over Poland. Many of these individuals soon became young researchers. I did not realize this at the time but a number of these individuals were ready for serious research work in Computer Science. The most advanced among these students and researchers was Witold Lipski (unfortunately died early - we have today the annual Lipski Competition for the brightest Polish young researcher in Computer Science). Zdzisław and I worked with Lipski who very quickly wrote a Ph.D. dissertation on information storage and retrieval systems, became a well-known database theorist and combinatorist. I will discuss how the combinatorics came into the picture in the next section. The most important aspect of that work was that very quickly we had in Warsaw a large group of young researchers working on databases and combinatorics. Soon, Lipski had a number of collaborators, both in database theory and combinatorics. In the next section. I will discuss how these two areas related in our work.

The interest in databases and their query languages came naturally to Pawlak. For a number of years, Zdzisław collaborated with a number of physicians. There were no subspecialty of Medical Informatics at the time, and one needed a vision to see that Medicine will be revolutionized by the computer applications. Nobody (at least in Warsaw) could imagine databases of medical cases except, of course, Zdzisław. He realized the potential of storing the medical data in databases, and more importantly, data mining the data so stored.

The formal descriptions of databases, query languages, and the possibility of testing formal properties of such databases were a major driving force of his considerations. And of course this projected on the work of the group of people around him. An important aspect (I will discuss it in more detail below) was that the (anonymized) records stored in databases did not form a set, but rather a bag (the inventors of relational model were, originally, against such approach, but today it is widely accepted).

There were several consequences of such possibility (i.e., existence of undistinguishable objects). Namely it is possible, even likely, that the query language is inadequate to describe the answers to queries that the user would like to make. This is a common case in medical applications. Often physicians see the symptoms not the underlying causes, and the language expressing symptoms may be inadequate to describe the essence of the underlying medical problem.

The key insight that Zdzisław had was that given a description language (which results in database scheme) that language may be good enough only for the approximate description of the set of objects that interests the user. There is more than one situation that occurs here and I will discuss the reasons for approximations in Section 4.3. Going into studies of approximations immediately changed the perspective. Namely several new aspects arose. For instance: *What are the measures of approximation? What are the query languages adequate for specific measures of approximation? Can one eliminate some attributes without lowering the quality*

of approximation? and many other questions. Not surprisingly, the resulting theory of *Rough Sets* related to Logic, Universal Algebra, but also to various aspects of Statistics. This is the source of the popular Rough Sets Theory widely studied today throughout the world.

4.2 Information Storage and Retrieval Systems, Databases

So, what were those information storage and retrieval systems (*i.s.r.* for short), and how were they motivated? The general questions, namely *What is a database and what are formal properties of databases?* were not settled at the time. Today, the researchers of database theory think about databases as relational systems in the sense used by logicians [1]. Surely, since the database is supposed to be stored, the relations (often called tables) need to be finite. Therefore, the corresponding logical system that may be used to describe these relational systems is some form of *finite model theory*, a fragment of model theory first studied by Y. Gurevich. Thinking about databases as collections of relations (tables) was proposed by E.F. Codd of IBM and quickly gained acceptance, first among theoreticians and then also, by use of query languages such as SQL, with a wide community of users (to be fair to others, there are many alternative ways of thinking about databases). Moreover, in a couple of years, there were *implementations* of relational databases, and soon they became competitive in their performance with respect to the older, non-SQL, systems.

Prior to the relational model, database systems were based on so-called network model and on hierarchical model. These previous models could not be, really, explained to the user community since they involved understanding processing of data within database systems, in particular data structures such as linked or double-linked lists. Relational model thinking was *declarative*; for the first time the user was thinking about *what* information she wants to get out of the system, not *how* she wants to get it.

Coming back to i.s.r., it was defined as a relational system, but on a single table. Also, it was heavily influenced by logic (rather than relational algebra). Let me shortly describe what happens when logicians look at the databases. First, one has to have a language. In case of databases, if one has to study sets of records, those need to be described. For that reason, one needed a language. The language had means to introduce *descriptors*. For that reason, Pawlak proposed to have a collection (called A) of all descriptors of the system. Actually, SQL does precisely the same (although in a clearer way, by means of types of attributes that are always finite, since even types such as integer, or real, are in reality finite). Descriptors split into classes called attributes. A natural way to do this is by means of an equivalence relation on the set of descriptors. Such construction presupposes that for attributes a_1 and a_2 the descriptors of type a_1 and of a_2 are disjoint. This may appear to be limiting, but really is not. For if we have three dimensions of a box and measure the size then

to describe a red box 35 centimeter long, 36 centimeters wide and 20 centimeters high, we can use the record

$$\langle length : 35, width : 35, height : 20, color : red \rangle.$$

In this way, the length equal to 35cm and width equal to 35 are disambiguated.

So now, an i.s.r. is a relational system $\mathcal{S} = \langle X, A, R_I, U \rangle$ where X is a set of objects (think about records but not necessarily different), A is the set of descriptors, R_I an equivalence relation partitioning descriptors into *attributes*, and finally, U is a function assigning to each descriptor $d \in A$ a subset of X consisting of records with value d. Since the descriptors (like in our example) carried the information about the attribute to which they belonged, there was no ambiguity. For instance, $U(height : 20)$ was the set of (descriptions of) boxes that had the height equal to 20. This choice of definition was motivated by the concept of *inverted file*, a construction not taught today in database courses (but current at the time), in which one stores for some or all descriptors the set of identifiers of records with that descriptor.

Once we have descriptors, we can build a free Boolean Algebra over that set. It is natural to find the set of boxes with the length equal to 35 and width 35, we need to compute intersection (Boolean meet) of two sets: of objects with the *length* 35 and of objects with the *width* 35. To facilitate answers to such queries (*give me the set of all boxes of length 35 and width 35*), the i.s.r. had the syntactical category of *terms*. These, in today parlance of SQL, corresponded to the queries that the user can ask. There was an inductive definition of term and the evaluation function $\| \cdot \|$. This evaluation function did what (simple) SQL queries do: returned the bag of records satisfying the suitable boolean condition. So the SQL query: SELECT * FROM *boxes* WHERE *length* = 35 AND *width* = 35; would be written as $\|length : 35 \cdot width : 35\|$. The query language of i.s.r. was significantly weaker than that of (even quite simple) SQL, since there were no comparators; all that were expressible were boolean operations. We soon realized that this was a problem and added extensions that allowed for comparisons, but we never truly recognized that the comparators are important. Moreover, SQL allows for "hiding" values of some attributes by means of projection operator. This was not available in i.s.r. SQL treats the answers to all queries as tables and these tables may have different schemata. This again was not available in i.s.r. But there were some advantages, too. Specifically, terms offered a possibility of describing formulas - the properties of the system in its entirety. SQL systems did not offer (and still do not offer) such capabilities, namely imposing general integrity constraints on the system[1]. The i.s.r. research did not study directly the first issue, but studied the second one. To give an example of the issues, let us assume that there is an additional attribute color. The language of i.s.r. allowed to express the properties of the systems such as "All red boxes have length 35". Today's SQL systems do not offer the language for testing such integrity constraints, although the user can write a program testing for such properties using so-called embedded SQL.

[1] Of course, *some* integrity constraints can be declared in SQL, but generally, SQL limits the capabilities of the database designer to specify the integrity constraints.

The researchers of i.s.r. devoted a significant amount of attention to various aspects of possible implementation of such systems. While today storage is inexpensive and all sorts of data are collected with massive databases of immense size, the situation was different in the 1970s. Storage was expensive and processing was slower. This lead to two important research topics: first, decomposing databases so that they required less space and second, organizing data on disk so that answers to *some* queries were computed in a simple manner. To give the example of this second issue, if the records of boxes with the length equal to 35 form a segment in the underlying organization of data then the answer of the query $\|length : 35\|$ is simple and requires minimal number of accesses to the disk. While the issue of decomposition of data went, essentially, away (we no longer require our students to normalize the data "to death", and normal forms beyond the so-called third normal form are not taught), the issue of organization of data did not go away and we are still concerned with minimizing the number of accesses to the disk. The theory behind the organization of data is a well-established topic. It involves both combinatorics (here the mathematics comes into play) and data structures. The mathematical foundations of the technology of storage were based, at the time, on *interval graphs* ([3]) and on the theory of Boolean matrices with *consecutive* 1*s property*. Lipski and his collaborators (this included me ([4]), and generally was a subclass of Pawlak's research group) devoted a lot of attention to these issues. In the modern setting, today, the issue did not disappear. As the larger amount of data are stored and then processed, the issue of quick retrieval becomes even more important since moving the data through the network becomes a "chocking point". For that reason the researchers of so-called Cloud Computing pay significant attention to data organization.

The work on i.s.r. under the name of *information systems* concerned the group of researchers around Pawlak throughout 1970s and resulted in a large body of research, eventually leading to studies of rough sets that I will report in the next section. One legacy of that research that slowly gained an acceptance in the mainstream database community was that the records can have duplicates. In theoretical terms this means that the tables are bags of records, not sets of records. This was obvious to Pawlak and his collaborators, because the language of i.s.r. naturally admitted a situation where two different objects had exactly same descriptions (certainly a common situation in databases of medical cases - one of the main motivation of Pawlak). To sum up, the investigations of i.s.r. prepared the ground for future related research on rough sets which will be discussed in the next section.

4.3 Rough Sets

The issue of the inadequacy of formal description language to describe desired families of sets of objects plagued (and still plagues) Computer Science. The nature of human natural language is such that when there is no adequate definition of some concept, we can invent an appropriate definition "on the fly". That is, the natural language constantly invents new concepts and vocabularies. With the formalized

languages, for instance of predicate calculus, change of vocabulary is still possible, but with each change comes the change of semantics and, often, of processing algorithms. The question of the changing language used to describe i.s.r. concerned the researchers from the beginning. The formal means to describe the inadequacy of the language was, again, a certain natural equivalence relation that can be associated with a given i.s.r. $\mathcal{S}$. Namely, $\mathcal{S}$ induces an equivalence relation in the set X of objects. This indiscernibility relation $\sim$ is defined as follows: $x \sim_{\mathcal{S}} y$ if for all descriptors a, $x \in U(a) \equiv y \in U(a)$. We will drop the subscript $\mathcal{S}$ when the system $\mathcal{S}$ is fixed. Hence, $x \sim y$ holds when, from the point of view of the query language of $\mathcal{S}$, the objects x and y are undistinguishable. The equivalence classes (cosets) of $\sim$ are minimal units that the language of $\mathcal{S}$ allows to describe. Assuming there is finite number of equivalence classes of $\sim$ (one can theoretically think about infinite $\mathcal{S}$'s but these do not appear in reality), the subsets of X that are describable by means of the query language of $\mathcal{S}$ are *precisely* the unions of (finite collections) these equivalence classes. Let us call, for the lack of a better word, these equivalence classes *monads*. If monads can have more than one element, then we face the following dilemma: What to do if that family of descriptions is inadequate to the needs of the user? There are several situations where we see this inadequacy. First, there may be situations where the monads are too big - in reality the descriptions should be finer, but we do not have the language good enough to describe the differences. This is common situation in medicine. Physicians strive to have adequate description of the underlying biological system (the patient) in objective terms. But before such description can be found, the less precise descriptions in forms of symptoms experienced by the patient is all that is available. But the same symptoms may show up in different medical conditions. In fact, the discovery of objective values is the subject of what is commonly known as medical tests, and the process of differentiation of description is practiced in medicine under the name of *differential diagnosis*. So, in this situation, which we call situation *I* the available query language is inadequate. But there is also another situation where the query language is adequate, but the shortest description of a set of interest is too big. To see what happens in this second situation (situation *II*) let us observe that, in principle, the number of monads is proportional to the product of the sizes of cosets of the relation R_I of the i.s.r. $\mathcal{S}$. Every set-theoretical union of monads is describable, but such descriptions may be very long! The question that Pawlak asked was how to handle both situations. His idea, first described in the paper [6], and then elaborated in details in his book [8], was to use approximations. But what approximations? It turned, eventually, out that more than one concept is involved. The question of inadequacy of the language can be treated as follows: with every subset $Y \subseteq X$ we can assign two definable subsets of X. Namely, the *greatest* definable subset of Y and the *least* definable superset of Y. These sets are commonly denoted $\underline{Y}$ and $\overline{Y}$, respectively. Of course, $\underline{Y}$ is the union of all monads included in Y, while $\overline{Y}$ is the union of monads that have nonempty intersection with Y. Having the concepts of $\underline{Y}$ and $\overline{Y}$ allows to *measure* inadequacy of the language of i.s.r. to describe a set Y. A variety of measures is possible, for instance the ratio of sizes of $\underline{Y}$ and Y (which is defined whenever $Y \subseteq X$ is nonempty).

But there are other measures, too. For instance the ratio of upper approximation to the lower one or of the size of Y to its upper approximation.

If we take the minima of the measures described above over all nonempty subsets of X, we get adequacy measure for the language itself! In other words, approximations allow to measure inadequacy of the language.

We observe that the information theorists devoted a significant attention to this problem. Some of the proposals such as *minimum description length* can be found in [9]. Once one starts to *measure* adequacy, new problems come to mind: feature extraction, feature constriction etc. We then land in the world of *machine learning*.

As mentioned above, the language can be adequate to describe a set $Y \subseteq X$, but description may be too large! If this is the case then we would like to find approximations of the set Y using a weaker language than that available from $\mathcal{S}$. This situation, under the name of *attribute reduction* trades precision for conciseness. Namely, we are willing to accept a pair of imprecise, but concise descriptions instead of one precise but impossibly long description.

Generally, then we trade impossibility of *adequate* description (either because of nonexistence of such description, or inadequacy of such description because of its size) by moving to approximations. It turns out (as shown in [7]) that rough sets (Pawlak approximations) can be characterized (in some precise sense) as best possible approximations. It did not surprise us, as in many situations Pawlak's intuition turned out to be very strong and confirmed by adequate mathematics. It is also worth mentioning that while the presence of a i.s.r. (that provides the description language) is beneficial, it is not a necessary ingredient of the approximation – all we need for this is the indiscernibility relation $\sim$, the point of view commonly accepted by the rough sets researchers. This level of abstraction allows to tie (as done of many researchers) rough sets with the universal-algebraic concepts of Boolean algebras with operators [2] and also with finite topologies.

4.4 Conclusions

Zdzisław was a true renaissance man: with many interests besides computer science he produced artistic short movies, wrote poetry, but also did practical things. Whatever he did the same enthusiasm and pervasive optimism present in his scientific work demonstrated itself in his actions. I, of course, benefited when he decided to build a shower stall in my small cottage in the country. Certainly the contrast between an academician living in "ivory tower" and a mason with his bricklayer trowel could not be bigger. Among many passions Zdzisław had was antique restoration. Like everything he did, this passion was contagious. So, when he and I brought to my apartment in Warsaw a round table bought in a consignment store (and in obvious need of restoration) my family was not pleased, and I had a new occupation for few months. I recall discussing with Zdzisław heating water using solar energy (yes, this was in 1970s!) and other innovations.

I left Poland in tumultuous year 1982 and in 1983 settled in Lexington, KY. Occasionally, I looked at Rough Sets (I mentioned one of those revisits above, there were other returns to that area of research, as well) but focused on another area, called nonmonotonic logic. This area dealt with another inadequacy of commonsense logic: namely of tentative and defeasible conclusions. This article is not the place to tell the story of that research. But of course, I followed developments in Rough Sets theory and met Zdzisław both in Poland when it was again possible to visit after the revolutions of 1989 and during his visits in the States. In particular I went to Nashville, TN (not far away from Lexington, at least for American distances), when in 1995 when Zdzisław made an invited presentation for the ACM which resulted in next year of Zadeh prize in Soft Computing.

As I am looking back, one thing is certain: working with Zdzisław was more than just science, it was life to the fullest.

Acknowledgements. We thank Professor Andrzej Skowron for initiating this paper and for a number of suggestions for improvements.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Jonsson, B., Tarski, A.: Boolean algebras with operators. Am. J. Math. 73, 891–939 (1951)
3. Lekkerkerker, C., Boland, D.: Representation of finite graphs by a set of intervals on the real line. Fund. Math. 51, 45–64 (1962)
4. Lipski, W., Marek, W.: File Organization, an Application of Graph Theory. In: Loeckx, J. (ed.) Automata, Languages and Programming. LNCS, vol. 14, pp. 270–279. Springer, Heidelberg (1974)
5. Marek, W., Pawlak, Z.: Information storage and retrieval systems: Mathematical foundations. Theor. Comput. Sci. 1, 331–354 (1976)
6. Marek, W., Pawlak, Z.: Rough sets and information systems. Technical Report 441. Computer Science Institute of PAS, Warsaw (1981)
7. Marek, W., Truszczynski, M.: Contributions to the theory of rough sets. Fund. Inform. 39, 389–409 (1999)
8. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
9. Rissanen, J.: Information and Complexity in Statistical Modeling. Springer, Heidelberg (2007)

Chapter 5
jMAF - Dominance-Based Rough Set Data Analysis Framework

Jerzy Błaszczyński, Salvatore Greco, Benedetto Matarazzo, Roman Słowiński, and Marcin Szeląg

Abstract. We present a rough set data analysis software jMAF. It employs java Rough Set (jRS) library in which are implemented data analysis methods provided by the (variable consistency) Dominance-based Rough Set Approach (DRSA). The chapter also provides some basics of the DRSA and of its variable consistency extension.

Keywords: Dominance-based rough set approach (DRSA), ordinal classification with monotonicity constraints, jMAF software, decision rules, reducts, variable consistency.

5.1 Introduction

jMAF is a rough set data analysis software written in Java language and available online[1]. It makes use of java Rough Set (jRS) library. Unlike other existing rough

Jerzy Błaszczyński · Roman Słowiński · Marcin Szeląg
Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
e-mail: {jblaszczynski,rslowinski,mszelag}@cs.put.poznan.pl

Roman Słowiński
Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland

Salvatore Greco, Benedetto Matarazzo
Department of Economics and Business, University of Catania
Corso Italia, 55, 95129 Catania, Italy
e-mail: {salgreco,matarazz}@unict.it

[1] http://www.cs.put.poznan.pl/jblaszczynski/Site/jRS.html

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 185–209.
springerlink.com

set exploration systems, *for example*, RSES and RSESlib[2], jMAF and jRS library implement methods of data analysis provided by the Dominance-based Rough Set Approach (DRSA), and by its extended version, the Variable Consistency Dominance-based Rough Set Approach (VC-DRSA). In this chapter, we give some basics of these two approaches, together with an example of jMAF usage that is meant to instruct novice users.

5.2 Reminder on the Dominance-Based Rough Set Approach

Dominance-based Rough Set Approach (DRSA) has been proposed by Greco, Matarazzo and Słowiński [11, 12, 13, 14, 35]. DRSA extends rough set theory proposed by Pawlak [28, 29, 32] and follows the suggestion formulated by Słowiński in [34], towards reasoning about decision situations with background knowledge about ordinal evaluations of objects from a universe, and about monotonic relationships between these evaluations, *for example* "the larger the mass and the smaller the distance, the larger the gravity" or "the greater the debt of a firm, the greater its risk of failure". Precisely, the monotonic relationships are assumed between evaluation of objects on condition attributes and their assignment to decision classes. The monotonic relationships are also interpreted as monotonicity constraint, because the better the evaluation of an object, the better should be the decision class the object is assigned to. For this reason, classification problems of this kind are called ordinal classification problems with monotonicity constraints. Many real-world classification problems fall into this category [7]. Typical examples are multiple criteria sorting and decision under uncertainty, where the order of value sets of attributes corresponds to increasing or decreasing order of preference of a decision maker. In these decision problems, the condition attributes are called *criteria*. Some tutorial presentations of DRSA are available in [15, 16, 36, 38].

It is worth stressing, however, that DRSA can also be used in data analysis of non-ordinal problems, that is, problems with no background knowledge about ordinal evaluations of objects, after an easy pre-processing of the input data [5]. It then gives more concise decision rules than the usual induction techniques designed for non-ordinal classification, without recurring to a pre-discretization of numerical attributes.

Although DRSA is a general methodology for reasoning about data describing ordinal classification problems with monotonicity constraints, in this chapter, we shall use the vocabulary typical for multiple criteria classification (called also sorting) problems.

[2] http://alfa.mimuw.edu.pl/~rses/

5.2.1 Decision Table

Let us consider a decision table [28] including a finite universe of objects (solutions, alternatives, and actions) U evaluated on a finite set of condition attributes $F = \{f_1, \ldots, f_n\}$, and on a single decision attribute d.

Table 5.1. Exemplary decision table with evaluations of students

Student	f_1 - Mathematics	f_2 - Physics	f_3 - Literature	d - Overall Evaluation
$S1$	good	medium	bad	bad
$S2$	medium	medium	bad	medium
$S3$	medium	medium	medium	medium
$S4$	good	good	medium	good
$S5$	good	medium	good	good
$S6$	good	good	good	good
$S7$	bad	bad	bad	bad
$S8$	bad	bad	medium	bad

The set of the indices of attributes is denoted by $I = \{1, \ldots, n\}$. Without loss of generality, $f_i : U \to \Re$ for each $i \in I$, and, for all objects $x, y \in U$, $f_i(x) \geq f_i(y)$ means that "x is at least as good as y with respect to attribute i", which is denoted by $x \succeq_i y$. Therefore, it is supposed that $\succeq_i$ is a complete preorder, that is, a strongly complete and transitive binary relation, defined on U on the basis of quantitative and qualitative evaluations $f_i(\cdot)$. Furthermore, decision attribute d makes a partition of U into a finite number of decision classes, $\boldsymbol{Cl}$=$\{Cl_1, \ldots, Cl_m\}$, such that each $x \in U$ belongs to one and only one class $Cl_t = \{x \in U : d(x) = t\}$, $t = 1, \ldots, m$. It is assumed that the classes are preference ordered, that is, for all $r, s = 1, \ldots, m$, such that $r > s$, the objects from Cl_r are preferred to the objects from Cl_s. More formally, if $\succeq$ is a *comprehensive weak preference relation* on U, defined by the decision attribute d, that is, if for all $x, y \in U$, $x \succeq y$ reads "x is at least as good as y", then it is supposed that

$$[x \in Cl_r,\ y \in Cl_s,\ r > s] \Rightarrow x \succ y,$$

where $x \succ y$ means $x \succeq y$ and *not* $y \succeq x$.

The above assumptions are typical for consideration of an ordinal classification with monotonicity constraints (or multiple criteria sorting) problem. Indeed, the decision table characterized above, includes examples of ordinal classification which constitute an input *preference information* to be analyzed using DRSA.

The sets to be approximated are called *upward union* and *downward union* of decision classes, respectively:

$$Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s, \quad Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s, \quad t = 1, \ldots, m.$$

The statement $x \in Cl_t^{\geq}$ reads "x belongs to at least class Cl_t", while $x \in Cl_t^{\leq}$ reads "x belongs to at most class Cl_t". Let us remark that $Cl_1^{\geq} = Cl_m^{\leq} = U$, $Cl_m^{\geq}$=Cl_m and $Cl_1^{\leq}$=Cl_1. Furthermore, for t=2,...,m,

$$Cl_{t-1}^{\leq} = U - Cl_t^{\geq} \quad \text{and} \quad Cl_t^{\geq} = U - Cl_{t-1}^{\leq}.$$

5.2.2 Dominance Cones as Granules of Knowledge

The key idea of DRSA is representation (approximation) of upward and downward unions of decision classes by *granules of knowledge* generated by attributes being criteria. These granules are *dominance cones* in the attribute values space.

x *dominates* y with respect to set of attributes $P \subseteq F$ (shortly, x *P-dominates* y), denoted by xD_Py, if for every attribute $f_i \in P$, $f_i(x) \geq f_i(y)$. The relation of P-dominance is reflexive and transitive, that is, it is a partial preorder.

Given a set of attributes $P \subseteq F$ and $x \in U$, the granules of knowledge used for approximation in DRSA are:

- a set of objects dominating x, called *P-dominating set* or *positive dominance cone*,
  $D_P^+(x)$=$\{y \in U: yD_Px\}$,
- a set of objects dominated by x, called *P-dominated set* or *negative dominance cone*,
  $D_P^-(x)$=$\{y \in U: xD_Py\}$.

Let us recall that the *dominance principle* [7] requires that an object x dominating object y on all considered attributes (i.e., x having evaluations at least as good as y on all considered attributes) should also dominate y on the decision (i.e., x should be assigned to at least as good decision class as y). Objects satisfying the dominance principle are called *consistent* and those which are violating the dominance principle are called *inconsistent*.

5.2.3 Approximation of Ordered Decision Classes

The *P-lower approximation* of $Cl_t^{\geq}$, denoted by $\underline{P}(Cl_t^{\geq})$, and the *P-upper approximation* of $Cl_t^{\geq}$, denoted by $\overline{P}(Cl_t^{\geq})$, are defined as follows ($t = 2,...,m$):

$$\underline{P}(Cl_t^{\geq}) = \{x \in U : D_P^+(x) \subseteq Cl_t^{\geq}\},$$
$$\overline{P}(Cl_t^{\geq}) = \{x \in U : D_P^-(x) \cap Cl_t^{\geq} \neq \emptyset\}.$$

The lower approximation of union of classes $Cl_t^{\geq}$ is composed of such objects x that the positive dominance cone $D_P^+(x)$ in the condition attribute space, with the origin whose coordinates are those of object x, is included in this union. The upper

approximation of union of classes $Cl_t^{\geq}$ is composed of such objects x that the negative dominance cone $D_P^-(x)$ in the condition attribute space, with the origin whose coordinates are those of object x, has a non-empty intersection with this union.

Analogously, one can define the *P-lower approximation* and the *P-upper approximation* of $Cl_t^{\leq}$ as follows ($t = 1,...,m-1$):

$$\underline{P}(Cl_t^{\leq}) = \{x \in U : D_P^-(x) \subseteq Cl_t^{\leq}\},$$
$$\overline{P}(Cl_t^{\leq}) = \{x \in U : D_P^+(x) \cap Cl_t^{\leq} \neq \emptyset\}.$$

The lower approximation of union of classes $Cl_t^{\leq}$ is composed of such objects x that the negative dominance cone $D_P^-(x)$ in the condition attribute space, with the origin whose coordinates are those of object x, is included in this union. The upper approximation of union of classes $Cl_t^{\leq}$ is composed of such objects x that the positive dominance cone $D_P^+(x)$ in the condition attribute space, with the origin whose coordinates are those of object x, has a non-empty intersection with this union.

Remark that these definitions are analogous to the definition of lower and upper approximation in basic rough set theory. However, here, the dominance cones are used instead of indiscernibility granules, and moreover, the approximated sets are here unions of ordered classes instead of single classes.

The P-lower and P-upper approximations so defined satisfy the following *inclusion property*, for all $P \subseteq F$:

$$\underline{P}(Cl_t^{\geq}) \subseteq Cl_t^{\geq} \subseteq \overline{P}(Cl_t^{\geq}), \quad t = 2,\ldots,m,$$
$$\underline{P}(Cl_t^{\leq}) \subseteq Cl_t^{\leq} \subseteq \overline{P}(Cl_t^{\leq}), \quad t = 1,\ldots,m-1.$$

The P-lower and P-upper approximations of $Cl_t^{\geq}$ and $Cl_t^{\leq}$ have an important *complementarity property*, according to which,

$$\underline{P}(Cl_t^{\geq}) = U - \overline{P}(Cl_{t-1}^{\leq}) \text{ and } \overline{P}(Cl_t^{\geq}) = U - \underline{P}(Cl_{t-1}^{\leq}),\ t=2,...,m,$$

$$\underline{P}(Cl_t^{\leq}) = U - \overline{P}(Cl_{t+1}^{\geq}) \text{ and } \overline{P}(Cl_t^{\leq}) = U - \underline{P}(Cl_{t+1}^{\geq}),\ t=1,...,m-1.$$

The *P-boundary* of $Cl_t^{\geq}$ and $Cl_t^{\leq}$, denoted by $Bn_P(Cl_t^{\geq})$ and $Bn_P(Cl_t^{\leq})$, respectively, are defined as follows:

$$Bn_P(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq}) - \underline{P}(Cl_t^{\geq}), \quad t = 2,\ldots,m,$$
$$Bn_P(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq}) - \underline{P}(Cl_t^{\leq}), \quad t = 1,\ldots,m-1.$$

Due to the above complementarity property, $Bn_P(Cl_t^{\geq}) = Bn_P(Cl_{t-1}^{\leq})$, for $t = 2,...,m$.

5.2.4 Quality of Approximation

For every $P \subseteq F$, the *quality of approximation* of the ordinal classification $\boldsymbol{Cl}$ by a set of attributes P is defined as the ratio of the number of objects P-consistent with the

dominance principle and the number of all the objects in U. Since the P-consistent objects are those which do not belong to any P-boundary $Bn_P(Cl_t^{\geq})$, $t = 2,\ldots,m$, or $Bn_P(Cl_t^{\leq})$, $t = 1,\ldots,m-1$, the quality of approximation of the ordinal classification $\boldsymbol{Cl}$ by a set of attributes P, can be written as

$$\gamma_P(\boldsymbol{Cl}) = \frac{\left|U - \left(\bigcup_{t=2,\ldots,m} Bn_P(Cl_t^{\geq})\right)\right|}{|U|} = \frac{\left|U - \left(\bigcup_{t=1,\ldots,m-1} Bn_P(Cl_t^{\leq})\right)\right|}{|U|}.$$

$\gamma_P(\boldsymbol{Cl})$ can be seen as a degree of consistency of the objects from U, where P is the set of attributes being criteria and $\boldsymbol{Cl}$ is the considered ordinal classification.

Moreover, for every $P \subseteq F$, the *accuracy of approximation* of union of ordered classes $Cl_t^{\geq}$, $Cl_t^{\leq}$ by a set of attributes P is defined as the ratio of the number of objects belonging to P-lower approximation and P-upper approximation of the union. Accuracy of approximation $\alpha_P(Cl_t^{\geq})$, $\alpha_P(Cl_t^{\leq})$ can be written as

$$\alpha_P(Cl_t^{\geq}) = \frac{\left|\underline{P}(Cl_t^{\geq})\right|}{|\overline{P}(Cl_t^{\geq})|}, \qquad \alpha_P(Cl_t^{\leq}) = \frac{\left|\underline{P}(Cl_t^{\leq})\right|}{|\overline{P}(Cl_t^{\leq})|}.$$

5.2.5 Reduction of Attributes

Each minimal (with respect to inclusion) subset $P \subseteq F$ such that $\gamma_P(\boldsymbol{Cl}) = \gamma_F(\boldsymbol{Cl})$ is called a *relative reduct* of $\boldsymbol{Cl}$ and is denoted by $RED_{\boldsymbol{Cl}}^k$, where $k \in K$, and K is a finite set of all reducts. Other types of reducts have been defined in the context of basic rough set theory [28, 41]. The intersection of all reducts is called the *core*, and is denoted by $CORE_{\boldsymbol{Cl}}$. Attributes in $CORE_{\boldsymbol{Cl}}$ cannot be removed from consideration without deteriorating the quality of approximation. This means that, in set F, there are three categories of attributes:

- *indispensable* attributes included in the core,
- *exchangeable* attributes included in some reducts, but not in the core,
- *redundant* attributes, neither indispensable nor exchangeable, and thus not included in any reduct.

An algorithm for reduction of attributes in the framework of the Dominance-based Rough Set Approach has been proposed in [42]. This algorithm has been implemented in jMAF.

5.2.6 Decision Rules

The dominance-based rough approximations of upward and downward unions of decision classes can serve to induce a generalized description of objects in terms of

"*if* ..., *then* ..." decision rules. For a given upward or downward union of classes, $Cl_t^{\geq}$ or $Cl_s^{\leq}$, the decision rules induced under a hypothesis that objects belonging to $\underline{P}(Cl_t^{\geq})$ or $\underline{P}(Cl_s^{\leq})$ are positive examples, and all the others are negative, suggest a *certain* assignment to "class Cl_t or better", or to "class Cl_s or worse", respectively. On the other hand, the decision rules induced under a hypothesis that objects belonging to $\overline{P}(Cl_t^{\geq})$ or $\overline{P}(Cl_s^{\leq})$ are positive examples, and all the others are negative, suggest a *possible* assignment to "class Cl_t or better", or to "class Cl_s or worse", respectively. Finally, the decision rules induced under a hypothesis that objects belonging to the intersection $\overline{P}(Cl_s^{\leq}) \cap \overline{P}(Cl_t^{\geq})$ are positive examples, and all the others are negative, suggest an *approximate* assignment to some classes between Cl_s and Cl_t $(s < t)$.

In the case of preference-ordered description of objects, set U is composed of examples of ordinal classification. Then, it is meaningful to consider the following five types of decision rules:

1) *certain* $\mathrm{D}_{\geq}$*-decision rules*, providing lower profile descriptions for objects belonging to $\underline{P}(Cl_t^{\geq})$:
if $f_{i_1}(x) \geq r_{i_1}$ *and* ... *and* $f_{i_p}(x) \geq r_{i_p}$, *then* $x \in Cl_t^{\geq}$,
where $\{i_1, \ldots, i_p\} \subseteq I$, $t = 2, \ldots, m$, $r_{i_1}, \ldots, r_{i_p} \in \Re$;

2) *possible* $\mathrm{D}_{\geq}$*-decision rules*, providing lower profile descriptions for objects belonging to $\overline{\underline{P}}(Cl_t^{\geq})$:
if $f_{i_1}(x) \geq r_{i_1}$ *and* ... *and* $f_{i_p}(x) \geq r_{i_p}$, *then* x *possibly belongs to* $Cl_t^{\geq}$,
where $\{i_1, \ldots, i_p\} \subseteq I$, $t = 2, \ldots, m$, $r_{i_1}, \ldots, r_{i_p} \in \Re$;

3) *certain* $\mathrm{D}_{\leq}$*-decision rules*, providing upper profile descriptions for objects belonging to $\underline{P}(Cl_t^{\leq})$:
if $f_{i_1}(x) \leq r_{i_1}$ *and* ... *and* $f_{i_p}(x) \leq r_{i_p}$, *then* $x \in Cl_t^{\leq}$,
where $\{i_1, \ldots, i_p\} \subseteq I$, $t = 1, \ldots, m-1$, $r_{i_1}, \ldots, r_{i_p} \in \Re$;

4) *possible* $\mathrm{D}_{\leq}$*-decision rules*, providing upper profile descriptions for objects belonging to $\overline{\underline{P}}(Cl_t^{\leq})$:
if $f_{i_1}(x) \leq r_{i_1}$ *and* ... *and* $f_{i_p}(x) \leq r_{i_p}$, *then* x *possibly belongs to* $Cl_t^{\leq}$,
where $\{i_1, \ldots, i_p\} \subseteq I$, $t = 1, \ldots, m-1$, $r_{i_1}, \ldots, r_{i_p} \in \Re$;

5) *approximate* $\mathrm{D}_{\geq\leq}$*-decision rules*, providing simultaneously lower and upper profile descriptions for objects belonging to $Cl_s \cup Cl_{s+1} \cup \ldots \cup Cl_t$, without possibility of discerning to which class:
if $f_{i_1}(x) \geq r_{i_1}$ *and* ... *and* $f_{i_k}(x) \geq r_{i_k}$ *and* $f_{i_{k+1}}(x) \leq r_{i_{k+1}}$ *and* ... *and* $f_{i_p}(x) \leq r_{i_p}$, *then* $x \in Cl_s \cup Cl_{s+1} \cup \ldots \cup Cl_t$,
where $\{i_1, \ldots, i_p\} \subseteq I$, $s, t \in \{1, \ldots, m\}$, $s < t$, $r_{i_1}, \ldots, r_{i_p} \in \Re$.

In the premise of a $\mathrm{D}_{\geq\leq}$-decision rule, there can be "$f_i(x) \geq r_i$" and "$f_i(x) \leq r_i'$", where $r_i \leq r_i'$, for the same $i \in I$. Moreover, if $r_i = r_i'$, the two conditions boil down to "$f_i(x) = r_i$".

Since a decision rule is a kind of implication, a *minimal* rule is understood as an implication such that there is no other implication with the premise of at least

the same weakness (in other words, a rule using a subset of elementary conditions and/or weaker elementary conditions) and the conclusion of at least the same strength (in other words, a $D_{\geq}$- or a $D_{\leq}$-decision rule assigning objects to the same union or sub-union of classes, or a $D_{\geq\leq}$-decision rule assigning objects to the same or smaller set of classes).

The rules of type 1) and 3) represent certain knowledge extracted from data (examples of ordinal classification), while the rules of type 2) and 4) represent possible knowledge; the rules of type 5) represent doubtful knowledge, because they are supported by inconsistent objects only.

Given a certain or possible $D_{\geq}$-decision rule $r \equiv$ "*if* $f_{i_1}(x) \geq r_{i_1}$ *and* ... *and* $f_{i_p}(x) \geq r_{i_p}$, *then* $x \in Cl_t^{\geq}$", an object $y \in U$ *supports* r if $f_{i_1}(y) \geq r_{i_1}$ *and* ... *and* $f_{i_p}(y) \geq r_{i_p}$. Moreover, object $y \in U$ supporting decision rule r is a *base* of r if $f_{i_1}(y) = r_{i_1}$ *and* ... *and* $f_{i_p}(y) = r_{i_p}$. Similar definitions hold for certain or possible $D_{\leq}$-decision rules and approximate $D_{\geq\leq}$-decision rules. A decision rule having at least one base is called *robust*. Identification of supporting objects and bases of robust rules is important for interpretation of the rules in multiple criteria decision analysis. The ratio of the number of objects supporting a rule and the number of all considered objects is called *relative support* of a rule. The relative support and the confidence ratio are basic characteristics of a rule; however, some *Bayesian confirmation measures* reflect much better the attractiveness of a rule [24]. In this sense one could consider a generalization of rough set approach in which approximations are defined taking into account confidence and also one or more confirmation measures. This idea constitutes the parameterized rough set approach proposed in [20].

A set of decision rules is *complete* if it covers all considered objects (examples of ordinal classification) in such a way that consistent objects are re-assigned to their original classes, and inconsistent objects are assigned to clusters of classes referring to this inconsistency. A set of decision rules is *minimal* if it is complete and non-redundant that is, exclusion of any rule from this set makes it incomplete. Remark that, in practice, using a complete set of rules is not always satisfactory because some specific rules with a weak support may negatively bias the classification of new objects.

Note that the syntax of decision rules induced from rough approximations defined using dominance cones is using consistently this type of granules. Each condition profile defines a dominance cone in n-dimensional condition space $\mathfrak{R}^n$, and each decision profile defines a dominance cone in one-dimensional decision space $\{1,\ldots,m\}$. In both cases, the cones are positive for $D_{\geq}$-rules and negative for $D_{\leq}$-rules. These decision rules are used to classify new objects according to a classification procedure described in [1].

Let us also remark that dominance cones corresponding to condition profiles can originate in any point of the condition attribute space $\mathfrak{R}^n$, identified with an object present in the decision table, without the risk of their being too specific. Thus, contrary to traditional granular computing, the condition space $\mathfrak{R}^n$ need not be discretized a priori. The cutting planes just pass through coordinates of objects from the decision table.

5.2.7 Variable Consistency Dominance-Based Rough Set Approaches

In DRSA, lower approximation of a union of ordered decision classes contains only consistent objects. Such a lower approximation is defined as a sum of dominance cones that are subsets of the approximated union. In practical applications, however, such a strong requirement may result in relatively small (and even empty) lower approximations. Therefore, several variants of DRSA have been proposed, relaxing the condition for inclusion of an object to the lower approximation. Variable Consistency Dominance-based Rough Set Approaches (VC-DRSA) include to lower approximations objects that are sufficiently consistent, according to different measures of consistency.

Although VC-DRSA is inspired by a similar approach proposed by Ziarko [45] and called VPRS, it is based on its own definitions of lower approximations which need to be monotonic in the sense described in [4].

Given a user-defined threshold value on a consistency measure, extended lower approximation of a union of classes is defined as a set of objects for which the consistency measure satisfies that constraint.

Several definitions of consistency measures have been considered in the literature so far. In the first papers concerning VC-DRSA [13, 23], consistency of objects has been calculated using rough membership measure [30, 44]. Then, in order to ensure monotonicity of lower approximation with respect to the dominance relation, some new consistency measures have been proposed and investigated in [2]. Recently, it has been observed that it is reasonable to require that a consistency measure used in the definition of the lower approximation satisfies a set of monotonicity properties [4]. Variable-consistency approaches involving such monotonic consistency measures are called Monotonic Variable Consistency Dominance-based Rough Set Approaches (Monotonic VC-DRSA) [3, 4].

Procedures for rule induction from dominance-based rough approximations obtained using VC-DRSA have been proposed in [6, 22]. These procedures are based on sequential covering procedure that was originally applied in the rough set context in the LEM algorithm [25].

5.3 Example of Application of jMAF

This section presents a didactic example which illustrates application of jMAF to an ordinal classification problem with monotonicity constraints. The surveys [15, 16, 17, 35, 36, 37, 38] include other examples of application of DRSA.

5.3.1 Running jMAF

You may find jMAF executable file in the location where you have unpacked the zip file that can be downloaded from http://www.cs.put.poznan.pl/jblaszczynski/Site/jRS.html. Please launch this file. A moment later you will see main jMAF window on your desktop. It should resemble the one presented in Figure 5.1.

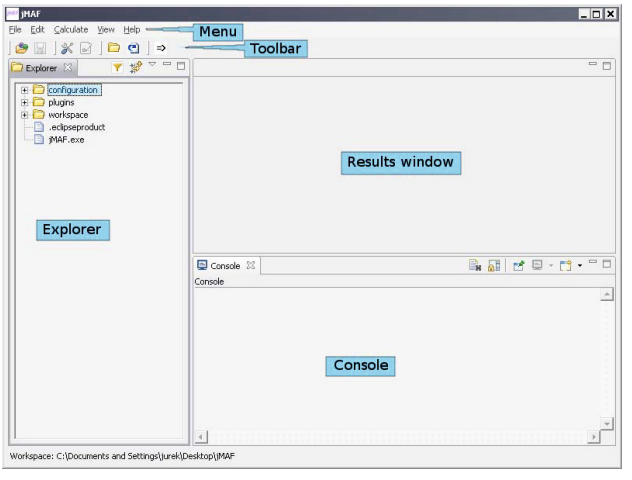


Fig. 5.1. jMAF main window

Now you have jMAF running in workspace folder located in the folder where it was launched from. You can check the content of workspace folder by examining the explorer window. The main jMAF window is divided into 4 sub windows: topmost menubar and toolbar, middle explorer and results window and bottom console window. There is also a status line at the bottom.

5.3.2 Decision Table

Let us consider the following ordinal classification problem. Students of a college must obtain an overall evaluation on the basis of their achievements in Mathematics, Physics, and Literature. These three subjects are clearly criteria (condition attributes), and the comprehensive evaluation is a decision attribute. For simplicity, the value sets of the attributes and of the decision attribute are the same, and they are composed of three values: bad, medium, and good. The preference order of these values is obvious. Thus, there are three preference-ordered decision classes, so the problem belongs to the category of ordinal classification. In order to build a preference model of the jury, DRSA is used to analyze a set of exemplary evaluations of students provided by the jury. These examples of ordinal classification constitute an input preference information presented as decision table in Table 5.2.

Note that the dominance principle obviously applies to the examples of ordinal classification, since an improvement of a student's score in one of three attributes with other scores unchanged should not worsen the student's overall evaluation but rather improve it.

Table 5.2. Exemplary decision table with evaluations of students (examples of ordinal classification)

Student	Mathematics	Physics	Literature	Overall Evaluation
$S1$	good	medium	bad	bad
$S2$	medium	medium	bad	medium
$S3$	medium	medium	medium	medium
$S4$	good	good	medium	good
$S5$	good	medium	good	good
$S6$	good	good	good	good
$S7$	bad	bad	bad	bad
$S8$	bad	bad	medium	bad

Observe that student $S1$ has not worse evaluations than student $S2$ on all the considered condition attributes; however, the overall evaluation of $S1$ is worse than the overall evaluation of $S2$. This violates the dominance principle so the two examples of ordinal classification, and only those, are inconsistent. One can expect that the quality of approximation of the ordinal classification represented by examples in Table 5.2 will be equal to 0.75.

5.3.3 Data File

As the first step, you should create a file containing data from the data table. You have now two choices-you may use spreadsheet-like editor or any plain text editor. For this example, we will focus on the second option.

Run any text editor that is available on your system installation. Enter the text shown below.

```
**ATTRIBUTES
+ Mathematics : [bad, medium, good]
+ Physics : [bad, medium, good]
+ Literature : [bad, medium, good]
+ Overall : [bad, medium, good]
decision: Overall

**PREFERENCES
Mathematics : gain
Physics : gain
```

```
Literature : gain
Overall : gain

**EXAMPLES
good medium bad bad
medium medium bad medium
medium medium medium medium
good good medium good
good medium good good
good good good good
bad bad bad bad
bad bad medium  bad

**END
```

Now, save the file as `students.isf` (for example in the jMAF folder). At this moment, you are able to open this file in jMAF.

5.3.4 Opening Data File

Use **File** | **Open** to open `students.isf` file. You will see a typical file open dialog. Please select your newly created file. Alternatively, you can double-click file in the explorer window if you have saved it in the workspace folder. If the file is not visible in explorer window, try right clicking on the explorer window and select from the context menu **Refresh** or **Switch workspace** to choose different workspace folder.

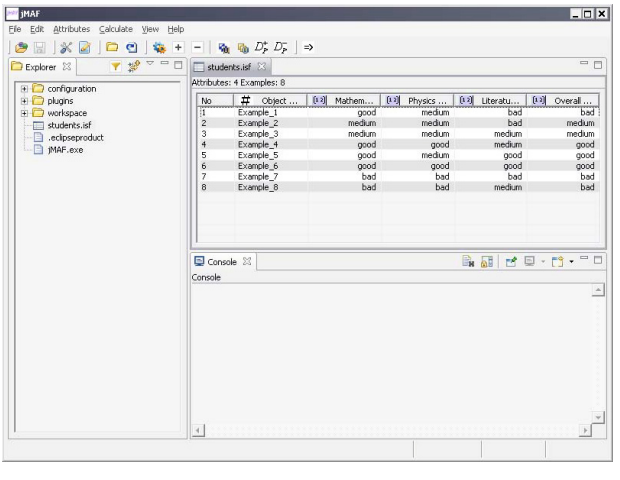


Fig. 5.2. File students.isf opened in jMAF

5.3.5 *Calculation of Dominance Cones*

One of the first steps of data analysis using rough set theory is calculation of dominance cones (P-dominating sets and P-dominated sets). To perform this step, you can select an example from the isf file in results window and use **Calculate** | **P-Dominance Sets** | **Calculate dominating set** or **Calculate** | **P-Dominance Sets** | **Calculate dominated set**. You can also use these options from the toolbar menu. The resulting dominance cones for student $S1$ are visible in Figures 5.3 and 5.4.

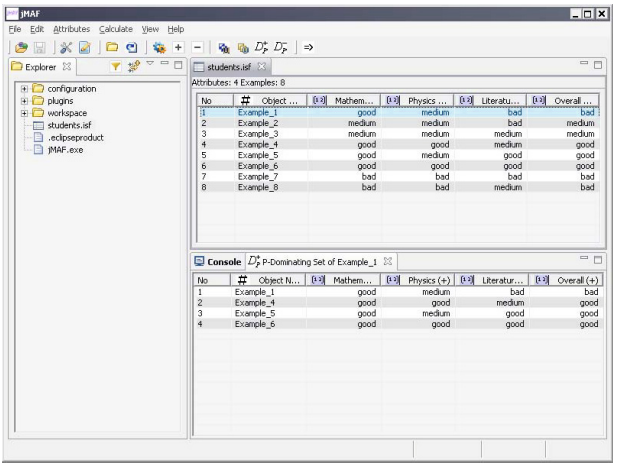


Fig. 5.3. P-dominating cone of Example 1

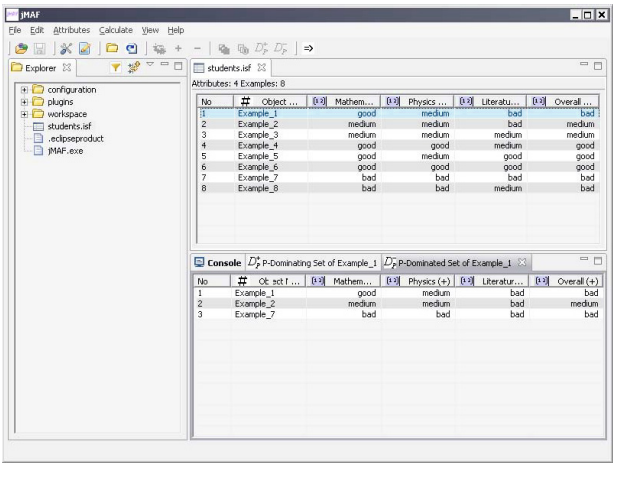


Fig. 5.4. P-dominated cone of Example 1

5.3.6 Calculation of Approximations

The next step in rough set analysis is calculation of approximations. Use **Calculate | Unions of classes | Standard unions of classes** to calculate DRSA unions and their approximations. Now, you should see an input dialog for calculation of approximations. It should look like the one presented in Figure 5.5.

Fig. 5.5. Input dialog for calculation of approximations

Leave default value of the consistency level parameter if you are looking for standard DRSA analysis. You can also set consistency level lower than one, to perform VC-DRSA analysis. The result would be that more of the objects from the upper approximations of unions with accuracy of approximation lower than one would be included in lower approximation. You should see the result as presented in Figure 5.6.

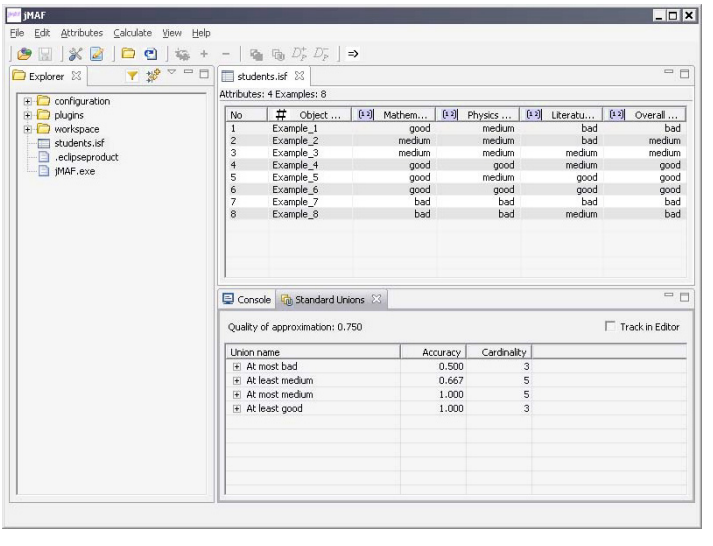


Fig. 5.6. Approximations of unions of classes

You can navigate in Standard Unions window to see more details concerning calculated approximations (they are presented in Figure 5.7).

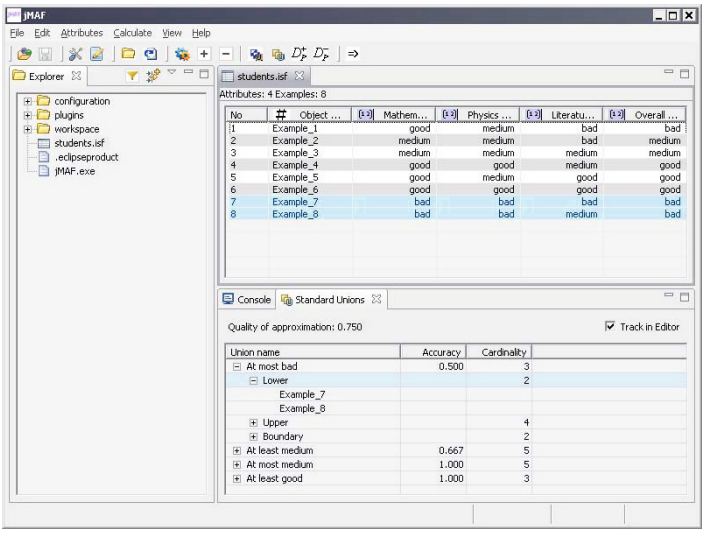


Fig. 5.7. Details of approximations of unions of classes

As you can see, quality of approximation equals 0.75, and accuracy of approximation in unions of classes ranges from 0.5 to 1.0. Lower approximation of union "at most" bad includes $S7$ and $S8$. Please select **Track in Editor** option to track your selection from Standard Unions window in the results window.

5.3.7 Calculation of Reducts

The list of all reducts can be obtained by selecting **Calculate** | **Reducts** | **All reducts**. As a result of this operation one can see all of reducts together with their cardinality, that is, number of criteria in a reduct. Additionally, the core of the calculated reducts is also shown (see Figure 5.8).

5.3.8 Induction of Decision Rules

Given the calculated in section 5.3.6 rough approximations, one can induce a set of decision rules representing the preferences of the jury. We will use one of the available methods - minimal covering rules (VC-DOMLEM algorithm).The idea is that evaluation profiles of students belonging to the lower approximations can serve

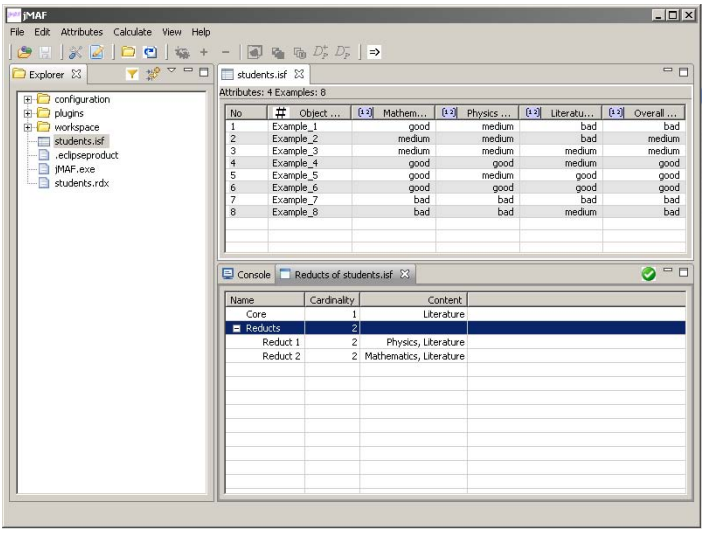


Fig. 5.8. List of calculated reducts and core

as a base for some certain rules, while evaluation profiles of students belonging to the boundaries can serve as a base for some approximate rules. In the example we will consider, however, only certain rules.

To induce rules use **Calculate** | **VC-DOMLEM algorithm**. You will see a dialog with parameters of rule induction that is presented in Figure 5.9. Leave default values of these parameters to perform standard rule induction for DRSA analysis.

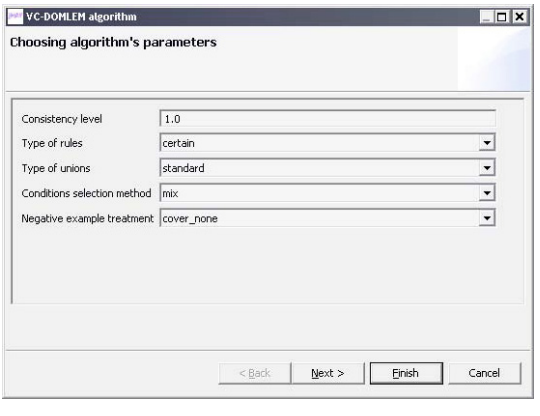


Fig. 5.9. Dialog with parameters of rule induction

To select where the result file with rules will be stored please edit output file in the following dialog (presented in Figure 5.10).

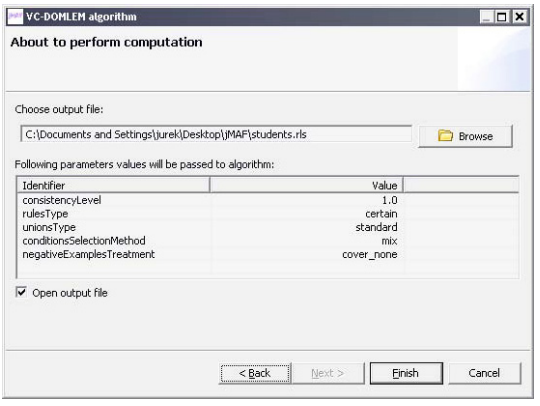


Fig. 5.10. Dialog with parameters of rule induction

The resulting rules are presented in results window (see Figure 5.11).

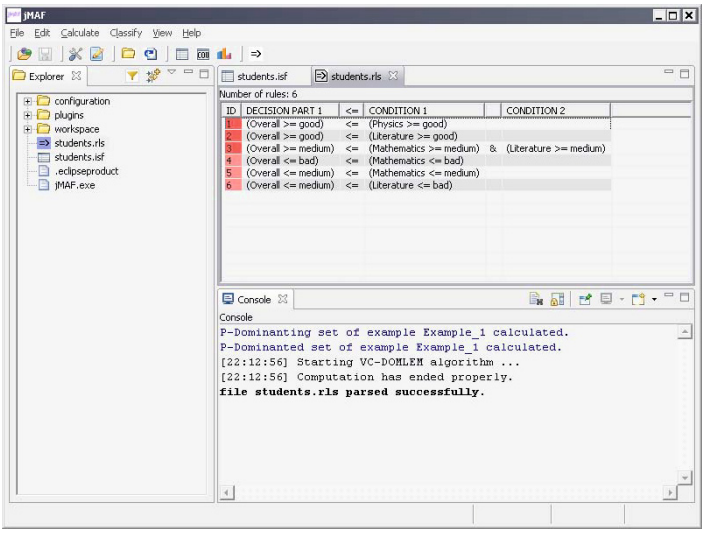


Fig. 5.11. Decision rules

Statistics of a rule selected in results window can be show by selecting **Open Statistics View associated with selected rule** from toolbar or from the context menu (right click on a rule). Statistics of the first rule are presented in Figure 5.12.

One can also see coverage of a rule (see Figure 5.13).

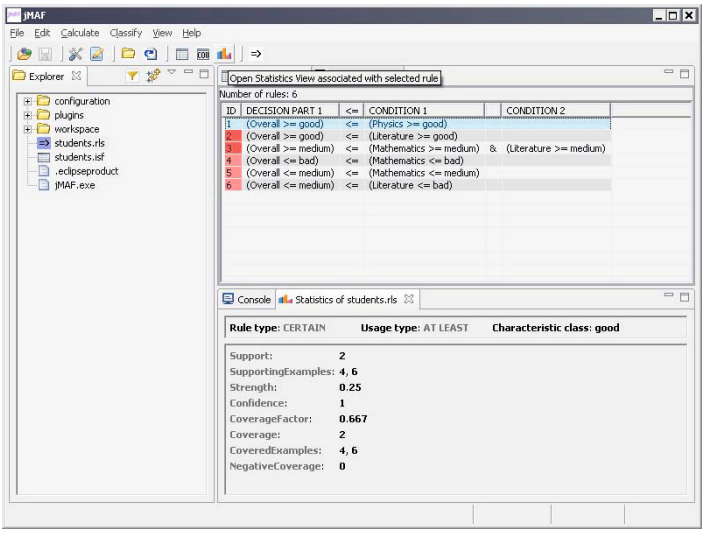


Fig. 5.12. Statistics of the first decision rule

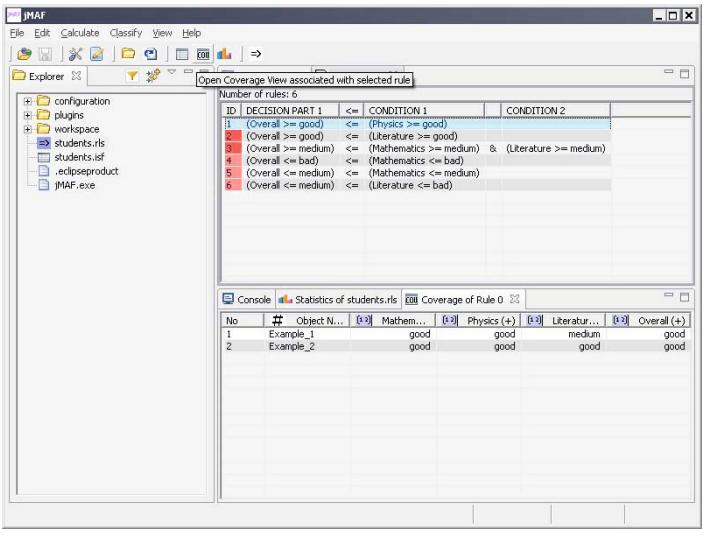


Fig. 5.13. Coverage of the first decision rule

5.3.9 Classification

Usually data analyst wants to know what is the value of induced rules, that is, how good they can classify objects. Thus, we proceed with an example of reclassification of learning data table for which rules were induced. To perform reclassification use

Classify | Reclassify learning examples. You will see a dialog with classification options. Select VCDRSA classification method as it is presented in Figure 5.14. Should you want to know more about VC-DRSA method, please see [1].

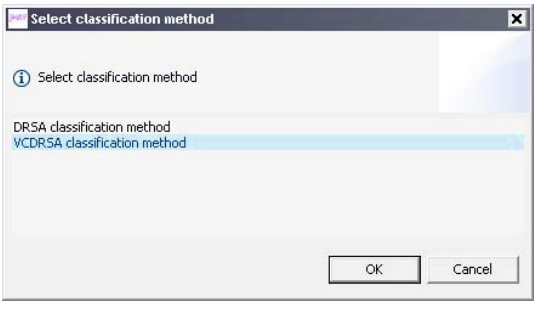


Fig. 5.14. Dialog with classification method

The results of classification are presented in a summary window as it is shown in Figure 5.15. Use **Details** button to see how particular objects were classified. The resulting window is presented in Figure 5.16. In this window, it is possible to see rules covering each of the classified examples and their classification.

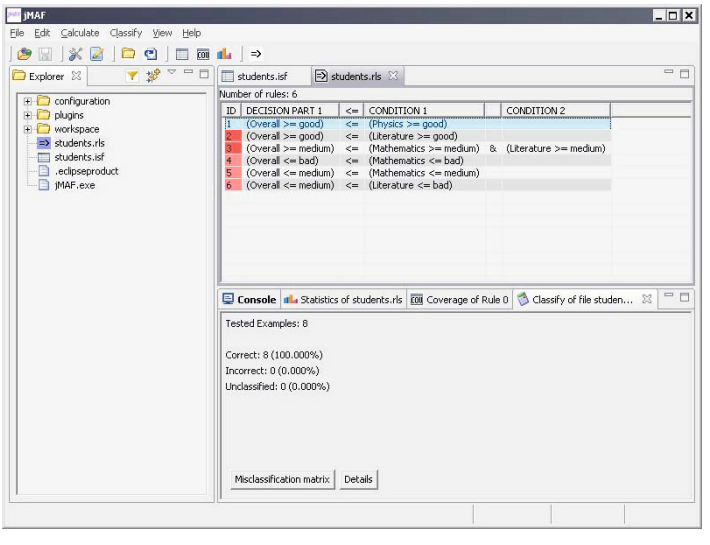


Fig. 5.15. Results of classification

Column “Certainty” in Fig. 5.16 refers to classification certainty score calculated in a way presented in [1].

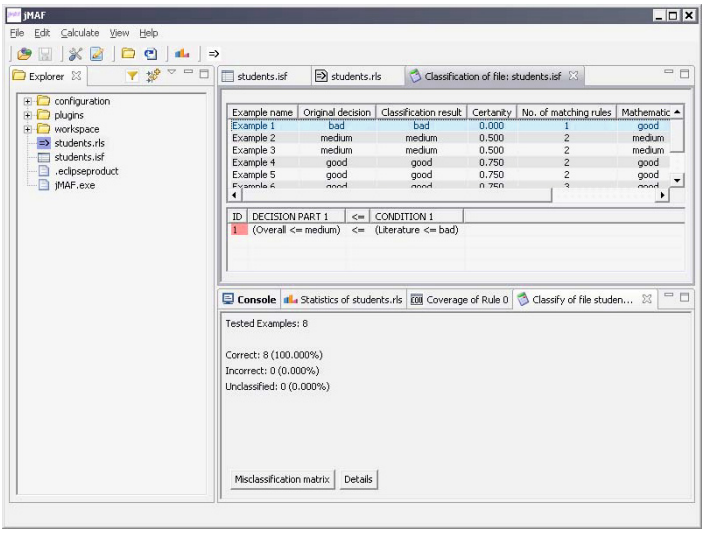


Fig. 5.16. Details of classification

5.4 Roadmap of Future Development of jMAF

jMAF is still under development. We plan to add the following functions to jMAF:

- induction of all decision rules with possibility of filtering a satisfactory subset of rules,
- handling of missing values,
- calculation of VC-reducts,
- measuring of Bayesian confirmation of attributes, and
- import of data files from popular spreadsheets.

5.5 Exemplary Applications of Dominance-Based Rough Set Approach

There are many possibilities of applying DRSA to real-life problems. The non-exhaustive list of potential applications includes:

- decision support in medicine: in this area, there are already many interesting applications (see, *e.g.*, [31, 26, 27, 43]); however, they exploit the classical rough set approach; applications requiring DRSA, which handle ordered value sets of medical signs, as well as monotonic relationships between the values of signs and the degree of a disease, are in progress;

- customer satisfaction survey: theoretical foundations for application of DRSA in this field are available in [18]; however, a fully documented application is still missing;
- bankruptcy risk evaluation: this is a field of many potential applications, as can be seen from promising results reported *for exmale* in [39, 40, 10]; however, a wider comparative study involving real data sets is needed;
- operational research problems, such as location, routing, scheduling or inventory management: these are problems formulated either in terms of classification of feasible solutions (see, *e.g.*, [9]) or in terms of interactive multiobjective optimization, for which there is a suitable IMO-DRSA [21] procedure;
- finance: this is a domain where DRSA for decision under uncertainty has to be combined with interactive multiobjective optimization using IMO-DRSA; some promising results in this direction have been presented in [19];
- ecology: assessment of the impact of human activity on the ecosystem is a challenging problem for which the presented methodology is suitable; the up-to-date applications are based on the classical rough set concept (see, *e.g.*, [33, 8]); however, it seems that DRSA handling ordinal data has a greater potential in this field.

5.6 Glossary

Multiple attribute (or multiple criteria) decision support aims at giving the decision maker (DM) a recommendation concerning a set of *objects U* (also called alternatives, actions, acts, solutions, options, candidates, etc.) evaluated from multiple points of view called *attributes* (also called features, variables, criteria, etc.).

Main categories of *multiple attribute (or multiple criteria) decision problems* are:

- *classification*, when the decision aims at assigning objects to predefined classes;
- *choice*, when the decision aims at selecting the best object;
- *ranking*, when the decision aims at ordering objects from the best to the worst.

Two kinds of *classification problems* are distinguished:

- *taxonomy*, when the value sets of attributes and the predefined classes are not preference ordered;
- *ordinal classification with monotonicity constraints* (also called *multiple criteria sorting*), when the value sets of attributes and the predefined classes are preference ordered, and there exist monotonic relationships between condition and decision attributes.

Two kinds of *choice problems* are distinguished:

- *discrete choice*, when the set of objects is finite and reasonably small to be listed;
- *multiple objective optimization*, when the set of objects is infinite and defined by constraints of a mathematical program.

If value sets of attributes are preference-ordered, they are called *criteria* or *objectives*, otherwise they keep the name of attributes.

Criterion is a real-valued function f_i defined on U, reflecting a worth of objects from a particular point of view, such that in order to compare any two objects $a, b \in U$ from this point of view, it is sufficient to compare two values: $f_i(a)$ and $f_i(b)$.

Dominance: object a is non-dominated in set U (Pareto-optimal) if and only if there is no other object b in U such that b is not worse than a in all considered criteria, and strictly better on at least one criterion.

Preference model is a representation of a value system of the decision maker on the set of objects with vector evaluations.

Rough set in universe U is an approximation of a set based on available information about objects of U. The rough approximation is composed of two ordinary sets, called *lower and upper approximation*. Lower approximation is a maximal subset of objects which, according to the available information, certainly belong to the approximated set, and upper approximation is a minimal subset of objects which, according to the available information, possibly belong to the approximated set. The difference between upper and lower approximation is called *boundary*.

Decision rule is a logical statement of the type "*if..., then...*", where the premise (condition part) specifies values assumed by one or more condition attributes and the conclusion (decision part) specifies an overall judgment.

References

1. Błaszczyński, J., Greco, S., Słowiński, R.: Multi-criteria classification – A new scheme for application of dominance-based decision rules. European Journal of Operational Research 181(3), 1030–1044 (2007)
2. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: On Variable Consistency Dominance-Based Rough Set Approaches. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 191–202. Springer, Heidelberg (2006)
3. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic Variable Consistency Rough Set Approaches. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 126–133. Springer, Heidelberg (2007)
4. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic variable consistency rough set approaches. International Journal of Approximate Reasoning 50(7), 979–999 (2009)
5. Błaszczyński, J., Greco, S., Słowiński, R.: Inductive discovery of laws using monotonic rules. Engineering Applications of Artificial Intelligence 25(2), 284–294 (2012)
6. Błaszczyński, J., Słowiński, R., Szeląg, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Information Sciences 181, 987–1002 (2011)
7. Figueira, J., Greco, S., Ehrgott, M. (eds.): Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, Berlin (2005)

8. Flinkman, M., Michałowski, W., Nilsson, S., Słowiński, R., Susmaga, R., Wilk, S.: Use of rough sets analysis to classify Siberian forest ecosystem according to net primary production of phytomass. INFOR 38, 145–161 (2000)
9. Gorsevski, P.V., Jankowski, P.: Discerning landslide susceptibility using rough sets. Computers, Environment and Urban Systems 32, 53–65 (2008)
10. Greco, S., Matarazzo, B., Słowiński, R.: A new rough set approach to evaluation of bankruptcy risk. In: Zopounidis, C. (ed.) Operational Tools in the Management of Financial Risks, pp. 121–136. Kluwer, Dordrecht (1998)
11. Greco, S., Matarazzo, B., Słowiński, R.: The use of rough sets and fuzzy sets. In: Gal, T., Stewart, T., Hanne, T. (eds.) Advances in Multiple Criteria Decision Making, ch. 14, pp. 14.1–14.59. Kluwer, Boston (1999)
12. Greco, S., Matarazzo, B., Słowiński, R.: Rough approximation of a preference relation by dominance relations. European J. Operational Research 117, 63–83 (1999)
13. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis, European J. of Operational Research 129, 1–47 (2001)
14. Greco, S., Matarazzo, B., Słowiński, R.: Multicriteria classification. In: Kloesgen, W., Żytkow, J. (eds.) Handbook of Data Mining and Knowledge Discovery, pp. 318–328. Oxford University Press (2002)
15. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach to Knowledge Discovery (I) - General Perspective (II) - Extensions and Applications. In: Zhong, N., Liu, J. (eds.) Intelligent Technologies for Information Analysis, ch. 20, 21, pp. 513–612. Springer, Berlin (2004)
16. Greco, S., Matarazzo, B., Słowiński, R.: Decision rule approach. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 507–563. Springer, Berlin (2005)
17. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-Based Rough Set Approach as a Proper Way of Handling Graduality in Rough Set Theory. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 36–52. Springer, Heidelberg (2007)
18. Greco, S., Matarazzo, B., Słowiński, R.: Customer satisfaction analysis based on rough set approach. Zeitschrift für Betriebswirtschaft 16(3), 325–339 (2007)
19. Greco, S., Matarazzo, B., Słowiński, R.: Financial portfolio decision analysis using Dominance-based Rough Set Approach. Invited paper at the 22nd European Conference on Operational Research (EURO XXII), August 8-11, Prague (2007)
20. Greco, S., Matarazzo, B., Słowiński, R.: Parameterized rough set model using rough membership and Bayesian confirmation measures. International Journal of Approximate Reasoning 49, 285–300 (2008)
21. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based rough set approach to interactive multiobjective optimization. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization: Interactive and Evolutionary Approaches, ch. 5. Springer, Berlin (2008)
22. Greco, S., Matarazzo, B., Słowiński, R., Stefanowski, J.: An Algorithm for Induction of Decision Rules Consistent with the Dominance Principle. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 304–313. Springer, Heidelberg (2001)
23. Greco, S., Matarazzo, B., Słowiński, R., Stefanowski, J.: Variable Consistency Model of Dominance-Based Rough Sets Approach. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 170–181. Springer, Heidelberg (2001)
24. Greco, S., Pawlak, Z., Słowiński, R.: Can Bayesian confirmation measures be useful for rough set decision rules? Engineering Applications of Artificial Intelligence 17, 345–361 (2004)

25. Grzymała-Busse, J.W.: LERS – A system for learning from examples based on rough sets. In: Słowiński, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
26. Michałowski, W., Rubin, S., Słowiński, R., Wilk, S.: Mobile clinical support system for pediatric emergencies. Journal of Decision Support Systems 36, 161–176 (2003)
27. Michałowski, W., Wilk, S., Farion, K., Pike, J., Rubin, S., Słowiński, R.: Development of a decision algorithm to support emergency triage of scrotal pain and its implementation in the MET system. INFOR 43, 287–301 (2005)
28. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
29. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
30. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R.R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 251–271. Wiley, New York (1994)
31. Pawlak, Z., Słowiński, K., Słowiński, R.: Rough classification of patients after highly selective vagotomy for duodenal ulcer. International Journal of Man-Machine Studies 24, 413–433 (1986)
32. Pawlak, Z., Słowiński, R.: Rough set approach to multi-attribute decision analysis. European J. of Operational Research 72, 443–459 (1994)
33. Rossi, L., Słowiński, R., Susmaga, R.: Rough set approach to evaluation of stormwater pollution. International Journal of Environment and Pollution 12, 232–250 (1999)
34. Słowiński, R.: Rough Set Learning of Preferential Attitude in Multi-criteria Decision Making. In: Komorowski, J., Raś, Z.W. (eds.) ISMIS 1993. LNCS (LNAI), vol. 689, pp. 642–651. Springer, Heidelberg (1993)
35. Słowiński, R., Greco, S., Matarazzo, B.: Rough Set Analysis of Preference-Ordered Data. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 44–59. Springer, Heidelberg (2002)
36. Słowiński, R., Greco, S., Matarazzo, B.: Rough set based decision support. In: Burke, E.K., Kendall, G. (eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, pp. 475–527. Springer, New York (2005)
37. Słowiński, R., Greco, S., Matarazzo, B.: Dominance-Based Rough Set Approach to Reasoning About Ordinal Data. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 5–11. Springer, Heidelberg (2007)
38. Słowiński, R., Greco, S., Matarazzo, B.: Rough sets in decision making. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7753–7786. Springer, New York (2009)
39. Słowiński, R., Zopounidis, C.: Application of the rough set approach to evaluation of bankruptcy risk. International Journal of Intelligent Systems in Accounting, Finance and Management 4, 27–41 (1995)
40. Słowiński, R., Zopounidis, C., Dimitras, A.I.: Prediction of company acquisition in Greece by means of the rough set approach. European Journal of Operational Research 100, 1–15 (1997)
41. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 43–84. Kluwer Academic Publishers, Dordrecht (1992)

42. Susmaga, R., Słowiński, R., Greco, S., Matarazzo, B.: Generation of reducts and rules in multi-attribute and multi-criteria classification. Control and Cybernetics 29(4), 969–988 (2000)
43. Wilk, S., Słowiński, R., Michałowski, W., Greco, S.: Supporting triage of children with abdominal pain in the emergency room. European Journal of Operational Research 160, 696–709 (2005)
44. Wong, S.K.M., Ziarko, W.: Comparison of the probabilistic approximate classification and the fuzzy set model. Fuzzy Sets and Systems 21(3), 357–362 (1987)
45. Ziarko, W.: Probabilistic approach to rough sets. International Journal of Approximate Reasoning 49(2), 272–284 (2008)

Chapter 6
Dynamic Programming Approach for Exact Decision Rule Optimization

Talha Amin, Igor Chikalov, Mikhail Moshkov, and Beata Zielosko

Abstract. This chapter is devoted to the study of an extension of dynamic programming approach that allows sequential optimization of exact decision rules relative to the length and coverage. It contains also results of experiments with decision tables from UCI Machine Learning Repository.

Keywords: Decision rules, dynamic programming, length, coverage.

6.1 Introduction

Decision rules are widely used as parts of classification algorithms (predictors), as a way for knowledge representation, and as parts of algorithms (parallel or nondeterministic) [13, 14, 16, 20].

There are different approaches to the construction of decision rules and reducts (decision rules can be considered as local reducts): brute-force approach that is applicable to tables with relatively small number of attributes, genetic algorithms [22, 24], Apriori algorithm [1], simulated annealing [9], Boolean reasoning [15, 17, 21], ant colony optimization [4, 10], algorithms based on decision tree construction [11, 14, 18], and different kinds of greedy algorithms [13, 15]. Each method can have different modifications. For example, as in the case of decision trees, we can use greedy algorithms based on different uncertainty measures (Gini index, entropy, etc.) to generate decision rules.

Talha Amin · Igor Chikalov · Mikhail Moshkov · Beata Zielosko
Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology, Thuwal 23955-6990, Saudi Arabia
e-mail:{talha.amin,igor.chikalov,mikhail.moshkov,beata.zielosko}
@kaust.edu.sa

Beata Zielosko
Institute of Computer Science, University of Silesia 39, Będzińska St.,
41-200 Sosnowiec, Poland

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 211–228.
springerlink.com

In many applications very often we are interested in the construction of short rules that cover many objects (we concentrate in this chapter on the consideration of exact rules). In particular, the choice of short rules is connected with the Minimum Description Length principle [19, 8] and Minimum Message Length [23]. The rule coverage is important to discover major patterns in the data. Unfortunately, the problems of minimization of length and maximization of coverage of decision rules are NP-hard.

In this chapter, we try to avoid this "restriction" for relatively small decision tables by the use of an extension of dynamic programming approach. If we can construct all subtables of the initial decision table given by systems of conditions of the kind "attribute = value" then we can describe the whole set of so-called irredundant decision rules, describe all irredundant rules with maximum coverage, and after that among these rules describe all rules with minimum length. We can change the order of optimization: describe all irredundant rules with minimum length, and after that describe among such rules all rules with maximum coverage. The obtained rules can be used both in usual [14] and online [5] classifiers.

We prove that by removal of some conditions from the left-hand side of each exact rule that is not irredundant we can obtain an irredundant (exact) decision rule whose length is at most the length of initial rule and the coverage is at least the coverage of initial rule. It means that we work not only with optimal rules among irredundant rules but also with optimal among all rules.

Similar approach to the decision tree optimization was considered in [2, 3, 6, 12]. First results for decision rules based on dynamic programming approach were obtained in [25]. The aim of this study was to find one decision rule with minimum length for each row.

In this chapter, we consider algorithms for optimization of irredundant decision rules relative to the length and coverage and results of experiments with some decision tables from UCI Machine Learning Repository [7] based on Dagger software system created in KAUST.

The chapter consists of eight sections. Section 6.2 is devoted to the consideration of irredundant decision rules. In Sect. 6.3, we study a directed acyclic graph that allows us to describe the whole set of irredundant decision rules. In Sect. 6.4, we consider a procedure of optimization of this graph (really, corresponding rules) relative to the length, and in Sect. 6.5 – relative to the coverage. In Sect. 6.6, we discuss possibilities of sequential optimization of rules relative to the length and coverage. Section 6.7 contains results of experiments with decision tables from UCI Machine Learning Repository, and Sect. 6.8 – conclusions.

6.2 Irredundant Decision Rules

First, we consider definitions of notions corresponding to decision tables and decision rules.

A *decision table* T is a rectangular table with n columns labeled with conditional attributes $f_1,\ldots,f_n$. Rows of this table are filled with nonnegative integers that are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer (decision) that is interpreted as a value of the decision attribute.

We denote by $N(T)$ the number of rows in the table T. The table T is called *degenerated* if T is empty (in this case $N(T)=0$) or all rows of T are labeled with the same decision.

A table obtained from T by the removal of some rows is called a *subtable* of the table T. Let T be nonempty, $f_{i_1},\ldots,f_{i_m}\in\{f_1,\ldots,f_n\}$ and $a_1,\ldots,a_m$ be nonnegative integers. We denote by $T(f_{i_1},a_1)\ldots(f_{i_m},a_m)$ the subtable of the table T which contains only rows that have numbers $a_1,\ldots,a_m$ at the intersection with columns $f_{i_1},\ldots,f_{i_m}$. Such nonempty subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ the set of attributes from $\{f_1,\ldots,f_n\}$ which are not constant on T. For any $f_i\in E(T)$, we denote by $E(T,f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i_1}=a_1\wedge\ldots\wedge f_{i_m}=a_m\rightarrow d \tag{6.1}$$

is called a *decision rule over* T if $f_{i_1},\ldots,f_{i_m}\in\{f_1,\ldots,f_n\}$, and $a_1,\ldots a_m,d$ are nonnegative integers. It's possible that $m=0$. In this case (6.1) is equal to the rule

$$\rightarrow d. \tag{6.2}$$

Let $r=(b_1,\ldots,b_n)$ be a row of T. We will say that the rule (6.1) is *realizable for* r, if $a_1=b_{i_1},\ldots,a_m=b_{i_m}$. If $m=0$ then the rule (6.2) is realizable for any row from T.

We will say that the rule (6.1) is *true for* T if each row of T for which the rule (6.1) is realizable has the decision d attached to it. Note that (6.1) is true for T if and only if the table $T'=T(f_{i_1},a_1)\ldots(f_{i_m},a_m)$ is degenerated and each row of T' is labeled with the decision d. If $m=0$ then the rule (6.2) is true for T if and only if T is degenerated and each row of T is labeled with the decision d.

If the rule (6.1) is true for T and realizable for r, we will say that (6.1) is a *decision rule for* T *and* r.

We will say that the rule (6.1) with $m>0$ is an *irredundant* decision rule for T and r if (6.1) is a decision rule for T and r and the following conditions hold:

(i) $f_{i_1}\in E(T)$ and if $m>1$ then $f_{i_j}\in E(T(f_{i_1},a_1)\ldots(f_{i_{j-1}},a_{j-1}))$ for $j=2,\ldots,m$;
(ii) if $m=1$ then the table T is nondegenerated, and if $m>1$ then the table $T(f_{i_1},a_1)\ldots(f_{i_{m-1}},a_{m-1})$ is nondegenerated.

If $m=0$ then the rule (6.2) is an *irredundant* decision rule for T and r if (6.2) is a decision rule for T and r, that is, if T is degenerated and each row of T is labeled with the decision d.

Lemma 6.1. *Let T be a nondegenerated decision table, $f_{i_1} \in E(T)$, $a_1 \in E(T, f_{i_1})$, and r be a row of the table $T' = T(f_{i_1}, a_1)$. Then the rule (6.1) with $m \geq 1$ is an irredundant decision rule for T and r if and only if the rule*

$$f_{i_2} = a_2 \wedge \ldots \wedge f_{i_m} = a_m \rightarrow d \tag{6.3}$$

is an irredundant decision rule for T' and r (if $m = 1$ then (6.3) is equal to $\rightarrow d$).

Proof. It is clear that (6.1) is a decision rule for T and r if and only if (6.3) is a decision rule for T' and r.

It's easy to show that the statement of lemma holds if $m = 1$. Let now $m > 1$.

Let (6.1) be an irredundant decision rule for T and r. Then from (i) it follows that $f_2 \in E(T')$ and if $m > 2$ then, for $j = 3, \ldots, m$, $f_{i_j} \in E(T'(f_{i_2}, a_2) \ldots (f_{i_{j-1}}, a_{j-1}))$. From (ii) it follows that T' is nondegenerated if $m = 2$ and $T'(f_{i_2}, a_1) \ldots (f_{i_{m-1}}, a_{m-1})$ is nondegenerated if $m > 2$. Therefore (6.3) is an irredundant decision rule for T' and r.

Let (6.3) be an irredundant decision rule for T' and r. Then, for $j = 2, \ldots, m$, $f_{i_j} \in E(T(f_{i_1}, a_1) \ldots (f_{i_{j-1}}, a_{j-1}))$. Also we know that $f_{i_1} \in E(T)$. Therefore the condition (i) holds. Since (6.3) is an irredundant decision rule for T' and r, we have $T(f_{i_1}, a_1)$ is nondegenerated if $m = 2$ and $T(f_{i_1}, a_1) \ldots (f_{i_{m-1}}, a_{m-1})$ is nondegenerated if $m > 2$. Therefore the condition (ii) holds, and (6.1) is an irredundant decision rule for T and r. □

Let τ be a decision rule over T and τ be equal to (6.1).

The number m of conditions on the left-hand side of τ is called the *length* of this rule and is denoted by $l(\tau)$. The length of decision rule (6.2) is equal to 0.

The *coverage* of τ is the number of rows in T for which τ is realizable and which are labeled with the decision d. We denote it by $c(\tau)$. The coverage of decision rule (6.2) is equal to the number of rows in T which are labeled with the decision d. If τ is true for T then $c(\tau) = N(T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m))$.

Proposition 6.1. *Let T be a nonempty decision table, r be a row of T and τ be a decision rule for T and r which is not an irredundant decision rule for T and r. Then by removal of some conditions from the left-hand side of τ we can obtain an irredundant decision rule $irr(\tau)$ for T and r such that $l(irr(\tau)) \leq l(\tau)$ and $c(irr(\tau)) \geq c(\tau)$.*

Proof. Let τ be equal to (6.1). Let T be degenerated. One can show that the rule $\rightarrow d$ is an irredundant decision rule for T and r. We denote this rule by $irr(\tau)$. It is clear that $l(irr(\tau)) \leq l(\tau)$ and $c(irr(\tau)) \geq c(\tau)$.

Let T be nondegenerated, and t be the minimum number from $\{1, \ldots, m\}$ such that $T(f_{i_1}, a_1) \ldots (f_{i_t}, a_t)$ is degenerated. If $t < m$ then we remove from τ the conditions $f_{i_{t+1}} = a_{t+1}, \ldots, f_{i_m} = a_m$. We denote the obtained rule by τ'. It is clear that τ' is a decision rule for T and r. If $f_{i_1} \notin E(T)$ then we remove the condition $f_{i_1} = a_1$ from τ'. For any $j \in \{2, \ldots, t\}$, if $f_{i_j} \notin E(T(f_{i_1}, a_1) \ldots (f_{i_{j-1}}, a_{j-1}))$ then we remove the condition $f_{i_j} = a_j$ from the left-hand side of the rule τ'.

One can show that the obtained rule is an irredundant decision rule for T and r. We denote it by $irr(\tau)$. It is clear that $l(\tau) \geq l(irr(\tau))$ and $c(\tau) \leq c(irr(\tau))$. □

6.3 Directed Acyclic Graph $\Delta(T)$

Now, we consider an algorithm that constructs a directed acyclic graph $\Delta(T)$ which will be used to describe the set of irredundant decision rules for T and for each row r of T. Nodes of the graph are some separable subtables of the table T. During each step, the algorithm processes one node and marks it with the symbol *. At the first step, the algorithm constructs a graph containing a single node T which is not marked with *.

Let us assume that the algorithm has already performed p steps. We describe now the step $(p+1)$. If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $\Delta(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet. If Θ is degenerated, then mark Θ with the symbol * and go to the step $(p+2)$. Otherwise, for each $f_i \in E(\Theta)$, draw a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{b_1, \ldots, b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, b_1), \ldots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. We label each row r of Θ with the set of attributes $E_{\Delta(T)}(\Theta, r) = E(\Theta)$. Mark the node Θ with the symbol * and proceed to the step $(p+2)$.

The graph $\Delta(T)$ is a directed acyclic graph. A node of this graph will be called *terminal* if there are no edges leaving this node. Note that a node Θ of $\Delta(T)$ is terminal if and only if Θ is degenerated.

Later, we will describe procedures of optimization of the graph $\Delta(T)$ relative to the length and coverage. As a result we will obtain a graph G with the same sets of nodes and edges as in $\Delta(T)$. The only difference is that any row r of each nondegenerated table Θ from G is labeled with a nonempty set of attributes $E_G(\Theta, r) \subseteq E(\Theta)$ possibly different from $E(\Theta)$.

Now for each node Θ of G and for each row r of Θ we describe a set of decision rules $Rul_G(\Theta, r)$. Let Θ be a terminal node of G: Θ is a degenerated table in which each row is labeled with the same decision d. Then

$$Rul_G(\Theta, r) = \{\rightarrow d\}.$$

Let now Θ be a nonterminal node of G such that for each child Θ' of Θ and for each row r' of Θ' the set of rules $Rul(\Theta', r')$ is already defined. Let $r = (b_1, \ldots, b_n)$ be a row of Θ labeled with a decision d. For any $f_i \in E_G(\Theta, r)$, we define the set of rules $Rul_G(\Theta, r, f_i)$ as follows:

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \rightarrow d : \alpha \rightarrow d \in Rul_G(\Theta(f_i, b_i), r)\}.$$

Then

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i).$$

Theorem 6.1. *For any node Θ of $\Delta(T)$ and for any row r of Θ, the set $Rul_{\Delta(T)}(\Theta, r)$ is equal to the set of all irredundant decision rules for Θ and r.*

Proof. We will prove this statement by induction on nodes in $\Delta(T)$. Let Θ be a terminal node of $\Delta(T)$. Then each row r of Θ is labeled with the same decision d. One can show that the rule $\rightarrow d$ is the only rule which is irredundant for Θ and r. Therefore, the set $Rul_{\Delta(T)}(\Theta, r)$ is equal to the set of all irredundant decision rules for Θ and r.

Let Θ be a nonterminal node of $\Delta(T)$, and for each child of Θ, the statement of theorem holds. Let $r = (b_1, \ldots, b_n)$ be a row of Θ which is labeled with a decision d. Using Lemma 6.1, we obtain that $Rul_{\Delta(T)}(\Theta, r)$ contains only irredundant decision rules for Θ and r.

Let τ be an irredundant decision rule for Θ and r. Since Θ is nondegenerated, the left-hand side of τ is nonempty. Therefore, τ can be represented in the form $f_i = b_i \wedge \alpha \rightarrow d$, where $f_i \in E(\Theta)$. Using Lemma 6.1, we obtain $\alpha \rightarrow d$ is an irredundant decision rule for $\Theta(f_i, b_i)$ and r. Based on inductive hypothesis, we obtain that the rule $\alpha \rightarrow d$ belongs to the set $Rul_{\Delta(T)}(\Theta(f_i, b_i), r)$. Therefore, $\tau \in Rul(\Theta, r)$. □

To illustrate algorithm studied in this chapter, we consider simple decision table T_0 (see Table 6.1).

Table 6.1. Decision table T_0

$T_0 =$

	f_1	f_2	f_3	
r_1	1	1	1	1
r_2	1	0	0	1
r_3	0	0	0	2
r_4	1	1	0	1

Algorithm stops partitioning of a subtable Θ of T_0 if Θ is degenerated or empty. We denote $G = \Delta(T_0)$.

For each node Θ of the graph G and for each row r of Θ we describe the set $Rul_G(\Theta, r)$. We will move from terminal nodes of G to the node T_0. Terminal nodes of the graph G are Θ_1, Θ_2, Θ_4, Θ_6, Θ_7, and Θ_8, Θ_9. For these nodes,

$Rul_G(\Theta_1, r_3) = \{\rightarrow 2\}$;
$Rul_G(\Theta_2, r_1) = Rul_G(\Theta_2, r_2) = Rul_G(\Theta_2, r_4) = \{\rightarrow 1\}$;
$Rul_G(\Theta_4, r_1) = Rul_G(\Theta_4, r_4) = \{\rightarrow 1\}$;
$Rul_G(\Theta_6, r_1) = \{\rightarrow 1\}$;
$Rul_G(\Theta_7, r_2) = \{\rightarrow 1\}$;
$Rul_G(\Theta_8, r_2) = Rul_G(\Theta_8, r_4) = \{\rightarrow 1\}$;
$Rul_G(\Theta_9, r_4) = \{\rightarrow 1\}$;

Now we can describe the sets of rules attached to rows of Θ_3 and Θ_5. There are nonterminal nodes of G. For Θ_3 all children Θ_1 and Θ_7 are already treated, and we have

$Rul_G(\Theta_3, r_2) = \{f_1 = 1 \rightarrow 1\}$;
$Rul_G(\Theta_3, r_3) = \{f_1 = 0 \rightarrow 2\}$.

For Θ_5 all children Θ_1, Θ_3, Θ_8, and Θ_9 are already treated, and we have

$Rul_G(\Theta_5, r_2) = \{f_1 = 1 \rightarrow 1, f_2 = 0 \wedge f_1 = 1 \rightarrow 1\}$;

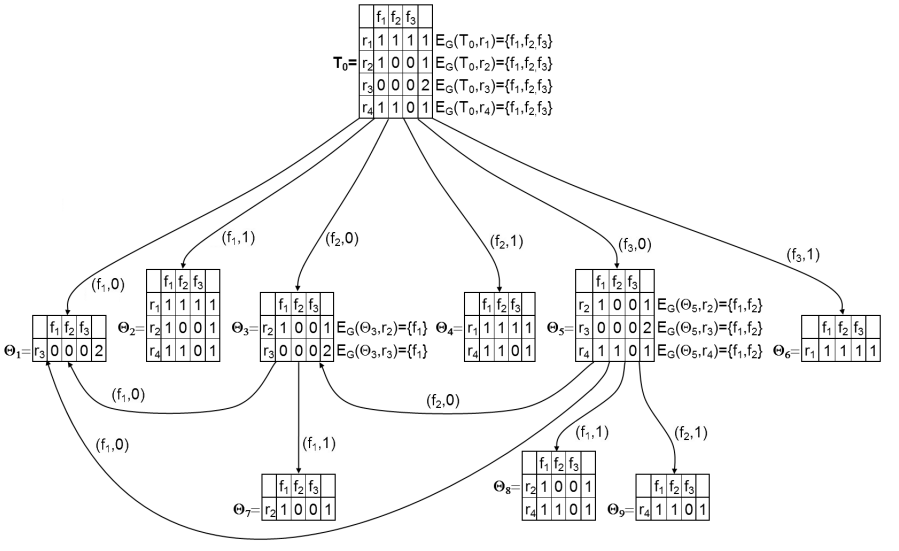


Fig. 6.1. Directed acyclic graph $G = \Delta(T_0)$

$Rul_G(\Theta_5, r_3) = \{f_1 = 0 \to 2, f_2 = 0 \wedge f_1 = 0 \to 2\}$;
$Rul_G(\Theta_5, r_4) = \{f_1 = 1 \to 1, f_2 = 1 \to 1\}$.

Finally, we can describe the sets of rules attached to rows of T_0:

$Rul_G(T_0, r_1) = \{f_1 = 1 \to 1, f_2 = 1 \to 1, f_3 = 1 \to 1\}$;
$Rul_G(T_0, r_2) = \{f_1 = 1 \to 1, f_2 = 0 \wedge f_1 = 1 \to 1, f_3 = 0 \wedge f_1 = 1 \to 1,$
$f_3 = 0 \wedge f_2 = 0 \wedge f_1 = 1 \to 1\}$;
$Rul_G(T_0, r_3) = \{f_1 = 0 \to 2, f_2 = 0 \wedge f_1 = 0 \to 2, f_3 = 0 \wedge f_1 = 0 \to 2,$
$f_3 = 0 \wedge f_2 = 0 \wedge f_1 = 0 \to 2\}$;
$Rul_G(T_0, r_4) = \{f_1 = 1 \to 1, f_2 = 1 \to 1, f_3 = 0 \wedge f_1 = 1 \to 1,$
$f_3 = 0 \wedge f_2 = 1 \to 1\}$.

6.4 Procedure of Optimization Relative to Length

We consider the procedure of optimization of the graph G relative to the length l. For each node Θ in the graph G, this procedure assigns to each row r of Θ the set $Rul^l_G(\Theta, r)$ of decision rules with minimum length from $Rul_G(\Theta, r)$ and the number $Opt^l_G(\Theta, r)$ – the minimum length of a decision rule from $Rul_G(\Theta, r)$.

The idea of the procedure is simple. It is clear that for each terminal node Θ of G and for each row r of Θ the following equalities hold:

$$Rul^l_G(\Theta, r) = Rul_G(\Theta, r) = \{\to d\},$$

where d is the decision attached to each row of Θ, and

$$Opt_G^l(\Theta, r) = 0.$$

Let Θ be a nonterminal node, $r = (b_1, \ldots, b_n)$ be a row of Θ labeled with a decision d. We know that

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i)$$

and, for $f_i \in E_G(\Theta, r)$,

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \to d : \alpha \to d \in Rul_G(\Theta(f_i, b_i), r)\}.$$

For $f_i \in E_G(\Theta, r)$, we denote by $Rul_G^l(\Theta, r, f_i)$ the set of all rules with the minimum length from $Rul_G(\Theta, r, f_i)$ and by $Opt_G^l(\Theta, r, f_i)$ – the minimum length of a decision rule from $Rul_G(\Theta, r, f_i)$.

One can show that

$$Rul_G^l(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \to d : \alpha \to d \in Rul_G^l(\Theta(f_i, b_i), r)\},$$

$$Opt_G^l(\Theta, r, f_i) = Opt_G^l(\Theta(f_i, b_i), r) + 1,$$

and

$$\begin{aligned} Opt_G^l(\Theta, r) &= \min\{Opt_G^l(\Theta, r, f_i) : f_i \in E_G(\Theta, r)\} \\ &= \min\{Opt_G^l(\Theta(f_i, b_i), r) + 1 : f_i \in E_G(\Theta, r)\}. \end{aligned}$$

It's easy to see also that

$$Rul_G^l(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r), Opt_G^l(\Theta(f_i, b_i), r) + 1 = Opt_G^l(\Theta, r)} Rul_G^l(\Theta, r, f_i).$$

We now describe the procedure of optimization of the graph G relative to the length l.

We will move from the terminal nodes of the graph G which are degenerated tables to the node T. We will assign to each row r of each table Θ the number $Opt_G^l(\Theta, r)$ which is the minimum length of a decision rule from $Rul_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in the nonterminal table Θ. We denote the obtained graph by G^l.

Let Θ be a terminal node of G. Then we assign to each row r of Θ the number $Opt_G^l(\Theta, r) = 0$.

Let Θ be a nonterminal node and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We assign the number

$$Opt_G^l(\Theta, r) = \min\{Opt_G^l(\Theta(f_i, b_i), r) + 1 : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G^l}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^l_G(\Theta(f_i, b_i), r) + 1 = Opt^l_G(\Theta, r)\}.$$

From the reasoning before the description of the procedure of optimization relative to the length the next statement follows.

Theorem 6.2. *For each node Θ of the graph G^l and for each row r of Θ the set $Rul_{G^l}(\Theta, r)$ is equal to the set $Rul^l_G(\Theta, r)$ of all rules with the minimum length from the set $Rul_G(\Theta, r)$.*

Figure 6.2 presents the directed acyclic graph G^l obtained from the graph G (see Fig. 6.1) by the procedure of optimization relative to the length.

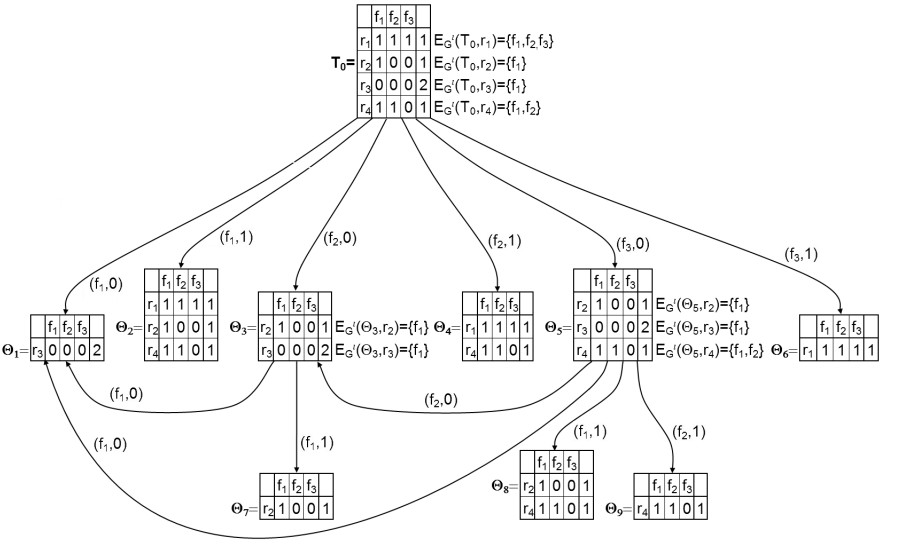


Fig. 6.2. Graph G^l

Using the graph G^l, can describe for each row r_i, $i = 1, \ldots, 4$, of the table T_0 the set $Rul^l_G(T_0, r_i)$ of all irredundant decision rules for T_0 and r_i with minimum length:
$Rul^l_G(T_0, r_1) = \{f_1 = 1 \to 1, f_2 = 1 \to 1, f_3 = 1 \to 1\}$;
$Rul^l_G(T_0, r_2) = \{f_1 = 1 \to 1\}$;
$Rul^l_G(T_0, r_3) = \{f_1 = 0 \to 2\}$;
$Rul^l_G(T_0, r_4) = \{f_1 = 1 \to 1, f_2 = 1 \to 1\}$.

6.5 Procedure of Optimization Relative to Coverage

We consider the procedure of optimization of the graph G relative to the coverage c. For each node Θ in the graph G, this procedure assigns to each row r of Θ the set $Rul_G^c(\Theta, r)$ of decision rules with maximum coverage from $Rul_G(\Theta, r)$ and the number $Opt_G^c(\Theta, r)$ – the maximum coverage of a decision rule from $Rul_G(\Theta, r)$.

The idea of the procedure is simple. It is clear that for each terminal node Θ of G and for each row r of Θ the following equalities hold:

$$Rul_G^c(\Theta, r) = Rul_G(\Theta, r) = \{\rightarrow d\},$$

where d is the decision attached to each row of Θ, and

$$Opt_G^c(\Theta, r) = N(\Theta).$$

Let Θ be a nonterminal node, $r = (b_1, \ldots, b_n)$ be a row of Θ labeled with a decision d. We know that

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i)$$

and, for $f_i \in E_G(\Theta, r)$,

$$Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \rightarrow d : \alpha \rightarrow d \in Rul_G(\Theta(f_i, b_i), r)\}.$$

For $f_i \in E_G(\Theta, r)$, we denote by $Rul_G^c(\Theta, r, f_i)$ the set of all rules with the maximum coverage from $Rul_G(\Theta, r, f_i)$ and by $Opt_G^c(\Theta, r, f_i)$ – the maximum coverage of a decision rule from $Rul_G(\Theta, r, f_i)$.

One can show that

$$Rul_G^c(\Theta, r, f_i) = \{f_i = b_i \wedge \alpha \rightarrow d : \alpha \rightarrow d \in Rul_G^c(\Theta(f_i, b_i), r)\},$$

$$Opt_G^c(\Theta, r, f_i) = Opt_G^c(\Theta(f_i, b_i), r),$$

and

$$\begin{aligned} Opt_G^c(\Theta, r) &= \max\{Opt_G^c(\Theta, r, f_i) : f_i \in E_G(\Theta, r)\} \\ &= \max\{Opt_G^c(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}. \end{aligned}$$

It's easy to see also that

$$Rul_G^c(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r), Opt_G^c(\Theta(f_i, b_i), r) = Opt_G^c(\Theta, r)} Rul_G^c(\Theta, r, f_i).$$

We now describe the procedure of optimization of the graph G relative to the coverage c.

We will move from the terminal nodes of the graph G which are degenerated tables to the node T. We will assign to each row r of each table Θ the number

$Opt^c_G(\Theta, r)$ which is the maximum coverage of a decision rule from $Rul_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in the nonterminal table Θ. We denote the obtained graph by G^c.

Let Θ be a terminal node of G. Then we assign the number

$$Opt^c_G(\Theta, r) = N(\Theta)$$

to each row r of Θ.

Let Θ be a nonterminal node and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We assign the number

$$Opt^c_G(\Theta, r) = \max\{Opt^c_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G^c}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^c_G(\Theta(f_i, b_i), r) = Opt^c_G(\Theta, r)\}.$$

From the reasoning before the description of the procedure of optimization relative to the coverage the next statement follows.

Theorem 6.3. *For each node Θ of the graph G^c and for each row r of Θ the set $Rul_{G^c}(\Theta, r)$ is equal to the set $Rul^c_G(\Theta, r)$ of all rules with the maximum coverage from the set $Rul_G(\Theta, r)$.*

Figure 6.3 presents the directed acyclic graph G^c obtained from the graph G (see Fig. 6.1) by the procedure of optimization relative to the coverage.

Using the graph G^c, can describe for each row r_i, $i = 1, \ldots, 4$, of the table T_0 the set $Rul^c_G(T_0, r_i)$ of all irredundant decision rules for T_0 and r_i with maximum coverage. We will give also the value $Opt^c_G(T_0, r_i)$ which is equal to the maximum coverage of an irredundant decision rule for T_0 and r_i. This value was obtained during the procedure of optimization of the graph G relative to the coverage. We have

$Rul^c_G(T_0, r_1) = \{f_1 = 1 \to 1\}$, $Opt^c_G(T_0, r_1) = 3$;
$Rul^c_G(T_0, r_2) = \{f_1 = 1 \to 1\}$, $Opt^c_G(T_0, r_2) = 3$;
$Rul^c_G(T_0, r_3) = \{f_1 = 0 \to 2, f_2 = 0 \wedge f_1 = 0 \to 2, f_3 = 0 \wedge f_1 = 0 \to 2,$
$f_3 = 0 \wedge f_2 = 0 \wedge f_1 = 0 \to 2\}$, $Opt^c_G(T_0, r_3) = 1$;
$Rul^c_G(T_0, r_4) = \{f_1 = 1 \to 1\}$, $Opt^c_G(T_0, r_4) = 3$.

6.6 Sequential Optimization

Theorems 6.2 and 6.3 show that we can make sequential optimization relative to the length and coverage. We can find all irredundant rules with maximum coverage, and after that among these rules find all rules with minimum length. We can also change the order of optimization: find all irredundant rules with minimum length, and after that find among such rules all rules with maximum coverage.

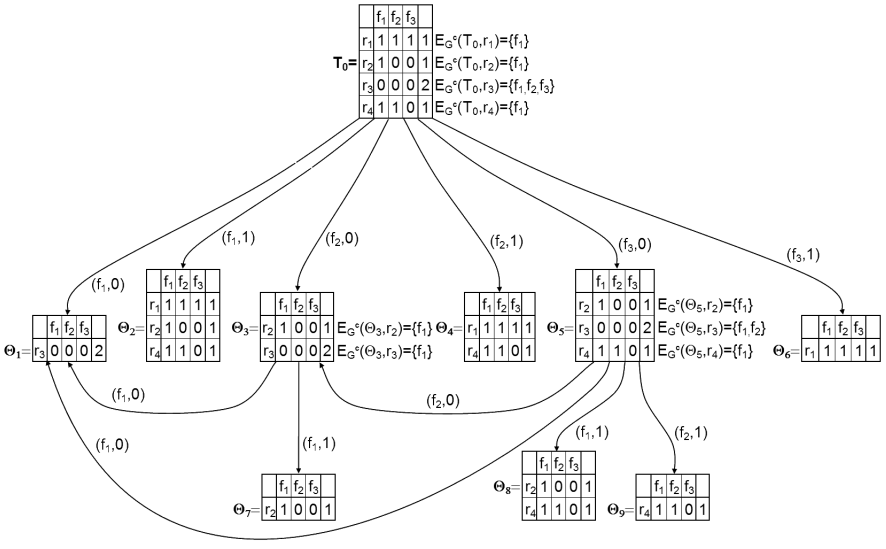


Fig. 6.3. Graph G^c

We will say that an irredundant decision rule for T and r is *totally optimal* if it has minimum length and maximum coverage among all irredundant decision rules for T and r. We can describe all totally optimal rules using the procedures of optimization relative to the length and coverage.

Set $G = \Delta(T)$. We apply the procedure of optimization relative to the coverage to the graph G. As a result we obtain the graph G^c and, for each row r of T – the value $Opt^c_G(T,r)$ which is equal to the maximum coverage of an irredundant decision rule for T and r.

Now we apply the procedure of optimization relative to the length to the graph G. As a result, we obtain the graph G^l. After that, we apply the procedure of optimization relative to the coverage to the graph G^l. As a result, we obtain the graph G^{lc} and, for each row r of T, – the value $Opt^c_{G^l}(T,r)$ which is equal to the maximum coverage of an irredundant decision rule for T and r among all irredundant decision rules for T and r with minimum length.

One can show that a totally optimal irredundant decision rule for T and r exists if and only if $Opt^c_G(T,r) = Opt^c_{G^l}(T,r)$. If the last equality holds then the set $Rul_{G^{lc}}(T,r)$ is equal to the set of all totally optimal irredundant decision rules for T and r.

It is clear that the results of sequential optimization of irredundant decision rules for T and r depend on the order of optimization (length+coverage or coverage+ length) if and only if there are no totally optimal irredundant decision rules for T and r.

Figure 6.4 presents the directed acyclic graph G^{lc} obtained from the graph G^l (see Fig. 6.2) by the procedure of optimization relative to the coverage.

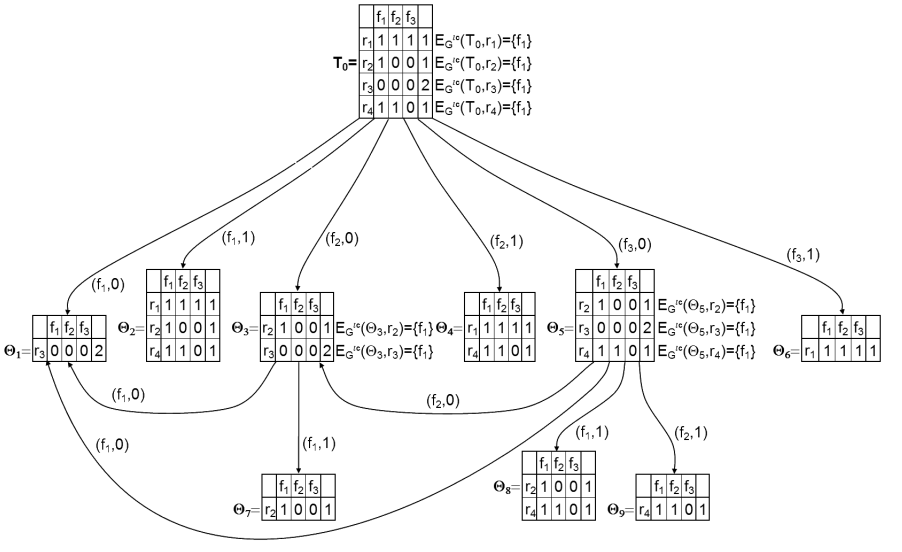


Fig. 6.4. Graph G^{lc}

Using the graph G^{lc}, can describe for each row r_i, $i = 1,\ldots,4$, of the table T_0 the set $Rul_{G^{lc}}(T_0, r_i)$ of all irredundant decision rules for T_0 and r_i which have maximum coverage among all irredundant decision rules for T_0 and r_i with minimum length. We will give also the value $Opt^c_{G^l}(T_0, r_i)$ which is equal to the maximum coverage of a decision rule for T_0 and r_i among all irredundant decision rules for T_0 and r_i with minimum length. This value was obtained during the procedure of optimization of the graph G^l relative to the coverage. We have

$$Rul_{G^{lc}}(T_0, r_1) = \{f_1 = 1 \to 1\}, Opt^c_{G^l}(T_0, r_1) = 3;$$
$$Rul_{G^{lc}}(T_0, r_2) = \{f_1 = 1 \to 1\}, Opt^c_{G^l}(T_0, r_2) = 3;$$
$$Rul_{G^{lc}}(T_0, r_3) = \{f_1 = 0 \to 2\}, Opt^c_{G^l}(T_0, r_3) = 1;$$
$$Rul_{G^{lc}}(T_0, r_4) = \{f_1 = 1 \to 1\}, Opt^c_{G^l}(T_0, r_4) = 3.$$

It's easy to see that $Opt^c_G(T_0, r_i) = Opt^c_{G^l}(T_0, r_i)$ for $i = 1,\ldots,4$. Therefore, for $i = 1,\ldots,4$, $Rul_{G^{lc}}(T_0, r_i)$ is the set of all totally optimal irredundant decision rules for T_0 and r_i.

6.7 Experimental Results

We considered a number of decision tables from UCI Machine Learning Repository [7]. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. In some tables there were equal rows with, possibly, different decisions. In this case each group of identical rows was replaced with a single row from the group with the most common decision for this group. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute.

For each of the considered decision tables T and for each row r of the table T, we find the minimum length of an irredundant decision rule for T and r, and the maximum coverage of an irredundant decision rule for T and r. After that, we find for rows of T the minimum length of a decision rule with minimum length, the maximum length of such a rule, and average length of rules with minimum length (one for each row). We find also for rows of T the minimum coverage of a decision rule with maximum coverage, the maximum coverage of such a rule, and the average coverage of rules with maximum coverage (one for each row). The considered results can be found in Table 6.2.

Table 6.2. Minimum length and maximum coverage of rules

Decision table	Number of rows	Number of cond. attr.	Min length of rules			Max coverage of rules		
			min	avg	max	min	avg	max
Adult-stretch	16	4	1	1.250	2	4	7.000	8
Agaricus-lepiota	8124	22	1	1.182	2	72	2135.463	2688
Balance-scale	625	4	3	3.197	4	1	4.213	5
Breast-cancer	266	9	1	2.665	6	1	9.534	25
Cars	1728	6	1	2.434	6	1	332.764	576
Flags	193	26	1	1.933	3	2	11.021	22
Hayes-roth-data	69	5	1	2.145	4	1	6.522	12
Hause-votes-84	279	16	2	2.538	5	2	73.523	97
Lenses	10	4	1	1.400	3	1	2.600	4
Lymphography	148	18	1	1.993	4	2	21.541	32
Monks-1-test	432	6	1	2.250	3	12	45.000	108
Monks-1-train	124	6	1	2.266	3	1	13.452	29
Monks-2-test	432	6	3	4.523	6	1	12.356	36
Monks-2-train	169	6	3	3.497	5	1	6.379	16
Monks-3-test	432	6	1	1.750	2	36	56.000	108
Monks-3-train	122	6	2	2.311	4	1	12.197	22
Mushrom	8124	22	1	1.182	2	72	2135.463	2688
Nursery	12960	8	1	3.118	8	1	1531.043	4320
Shuttle-landing-control	15	6	1	1.400	4	1	2.133	3
Soybean-small	47	35	1	1.000	1	10	12.532	17
Spect-test	169	22	1	1.479	8	2	58.047	67
Teeth	23	8	1	2.261	4	1	1.000	1
Tic-tac-toe	958	9	3	3.017	4	6	66.681	90
Zoo	59	16	1	1.559	4	3	11.068	19

Table 6.3. Totally optimal rules

Decision table	Number of rows	Number of rows with tot. opt. rules	Number of tot. opt. rules		
			min	avg	max
Adult-stretch	16	16	1	1.250	2
Agaricus-lepiota	8124	612	0	0.075	1
Balance-scale	625	625	1	1.637	4
Breast-cancer	266	133	0	0.650	5
Cars	1728	1728	1	1.115	2
Flags	193	53	0	0.311	4
Hayes-roth-data	69	69	1	1.145	2
Hause-votes-84	279	101	0	0.384	2
Lenses	10	10	1	1.200	2
Lymphography	148	52	0	0.358	2
Monks-1-test	432	432	1	1.000	1
Monks-1-train	124	120	0	1.016	2
Monks-2-test	432	432	1	1.111	3
Monks-2-train	169	157	0	1.112	3
Monks-3-test	432	432	1	1.056	2
Monks-3-train	122	122	1	1.164	3
Mushrom	8124	612	0	0.075	1
Nursery	12960	12960	1	1.140	4
Shuttle-landing-control	15	13	0	0.933	2
Soybean-small	47	37	0	1.851	5
Spect-test	169	108	0	0.645	2
Teeth	23	23	1	3.609	10
Tic-tac-toe	958	942	0	0.994	2
Zoo	59	44	0	1.305	17

Recall that an irredundant decision rule for T and r is called totally optimal if it has minimum length and maximum coverage among all irredundant decision rules for T and r. For each of the considered decision tables T, we count the number of rows r such that there exists a totally optimal irredundant decision rule for T and r. We find minimum, average and maximum number of totally optimal irredundant decision rules for T and r among all rows r of T (see Table 6.3).

In Table 6.4, we present results of sequential optimization of exact decision rules. For tables with rows that have no totally optimal rules, we make two steps of optimization – relative to the length and then relative to the coverage (column "length+coverage"). After that, we find average length and average coverage of rules after two steps of optimization.

We consider also the reversed order of optimization. For tables with rows that have no totally optimal rules, we make two steps of optimization – relative to the coverage and then relative to the length (column "coverage+length"). After that, we find average length and average coverage of rules after two steps of optimization.

We consider both average among all rows (column "avg_1") and average among all rows without totally optimal rules (column "avg_2"). Column "At" contains the number of conditional attributes, column "Row" contains the number of rows, and column "Row′" contains the number of rows without totally optimal rules in the considered decision table.

Table 6.4. Sequential optimization of exact decision rules

Decision table	At	Row	Row′	length+coverage				coverage+length			
				length		coverage		length		coverage	
				avg_1	avg_2	avg_1	avg_2	avg_1	avg_2	avg_1	avg_2
Agaricus-lepiota	22	8124	7512	1.182	1.193	1370.132	1309.919	2.514	2.633	2135.463	2137.601
Breast-cancer	9	266	133	2.665	2.376	7.038	4.429	3.429	3.902	9.534	9.421
Flags	26	193	140	1.933	1.943	6.394	3.429	4.119	4.957	11.021	9.807
Hause-votes-84	16	279	178	2.538	2.494	65.409	62.685	3.520	4.034	73.523	75.404
Lymphography	18	148	96	1.993	1.958	15.169	9.698	2.932	3.406	21.541	19.521
Monks-1-train	6	124	4	2.266	3.000	13.395	2.000	2.298	4.000	13.452	3.750
Monks-2-train	6	169	12	3.497	3.167	6.249	3.250	3.568	4.167	6.379	5.083
Mushrom	22	8124	7512	1.182	1.193	1370.132	1309.919	2.514	2.633	2135.463	2137.601
Shuttle-landing	6	15	2	1.400	1.500	1.867	1.000	1.733	4.000	2.133	3.000
Soybean-small	35	47	10	1.000	1.000	12.234	8.600	1.213	2.000	12.532	10.000
Spect-test	22	169	61	1.479	1.639	53.550	39.639	1.964	2.984	58.047	52.098
Tic-tac-toe	9	958	16	3.017	4.000	66.580	2.000	3.033	5.000	66.681	8.000
Zoo	16	59	15	1.559	2.200	10.525	3.200	1.881	3.467	11.068	5.333

6.8 Conclusions

We studied an extension of dynamic programming approach to the optimization of exact decision rules relative to the length and coverage. The considered approach allows us to describe the whole set of irredundant decision rules and optimize these rules sequentially relative to the length and coverage or relative to the coverage and length. We considered results of experiments with decision tables from UCI Machine Learning Repository [7]. Future study will be connected with the consideration of weighted length and weighted coverage of decision rules.

Acknowledgements. This research was supported by King Abdullah University of Science and Technology in the frameworks of joint project with Nizhni Novgorod State University "Novel Algorithms in Machine Learning and Computer Vision, and their High Performance Implementations", Russian Federal Program "Research and Development in Prioritized Directions of Scientific-Technological Complex of Russia in 2007–2013".

The authors are greatly indebted to Professor Andrzej Skowron for the comments and suggestions.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proc. 20th Int. Conf. on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann (1994)

2. Alkhalid, A., Chikalov, I., Moshkov, M.: On Algorithm for Building of Optimal α-Decision Trees. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 438–445. Springer, Heidelberg (2010)
3. Alkhalid, A., Chikalov, I., Hussain, S., Moshkov, M.: Extensions of dynamic programming as a new tool for decision tree optimization. In: S. Ramanna, R.J. Howlett, L.C. Jain (eds.) Emerging Paradigms in Machine Learning. Springer (to appear, 2012)
4. Boryczka, U., Kozak, J.: New algorithms for generation decision trees – Ant-Miner and its modifications. Foundations of Computational Intelligence 6, 229–262 (2009)
5. Chikalov, I., Moshkov, M., Zielosko, B.: Online Learning Algorithm for Ensemble of Decision Rules. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS (LNAI), vol. 6743, pp. 310–313. Springer, Heidelberg (2011)
6. Chikalov, I.V.: On Algorithm for Constructing of Decision Trees with Minimal Number of Nodes. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 139–143. Springer, Heidelberg (2001)
7. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010), http://archive.ics.uci.edu/ml (cited July 2, 2011)
8. Grünwald, P.D.: The Minimum Description Length Principle. Foreword by Jorma Rissanen. In: Adaptive Computation and Machine Learning. MIT Press (2007)
9. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Transactions on Knowledge and Data Engineering 16, 1457–1471 (2000)
10. Liu, B., Abbass, H.A., McKay, B.: Classification rule discovery with ant colony optimization. In: Proc. IEEE/WIC Int. Conf. on Intelligent Agent Technology, IAT 2003, pp. 83–88. IEEE Computer Society, Washington, DC (2003)
11. Michalski, S.R., Pietrzykowski, J.: iAQ: A program that discovers rules. AAAI-07 AI Video Competition. In: 22nd AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada (2007) http://videolectures.net/aaai07_michalski_iaq/ (cited February 27, 2012)
12. Moshkov, M., Chikalov, I.: On algorithm for constructing of decision trees with minimal depth. Fundamenta Informaticae 41, 295–299 (2000)
13. Moshkov, M., Piliszczuk, M., Zielosko, B.: Partial Covers, Reducts and Decision Rules in Rough Sets: Theory and Applications. SCI, vol. 145. Springer, Heidelberg (2008)
14. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning: A Rough Set Approach. SCI, vol. 360. Springer, Heidelberg (2011)
15. Nguyen, H.S.: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)
16. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
17. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177, 41–73 (2007)
18. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
19. Rissanen, J.: Modeling by shortest data description. Automatica 14, 465–471 (1978)
20. Skowron, A.: Rough sets in KDD. In: Shi, Z., Faltings, B., Musem, M. (eds.) 16th World Computer Congress. Proc. Conf. Intelligent Information Processing, pp. 1–17. Publishing House of Electronic Industry, Beijing (2000)

21. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
22. Ślęzak, D., Wróblewski, J.: Order-Based Genetic Algorithms for the Search of Approximate Entropy Reducts. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 308–311. Springer, Heidelberg (2003)
23. Wallace, C.S.: Statistical and Inductive Inference by Minimum Message Length. Information Science and Statistics. Springer, New York (2005)
24. Wróblewski, J.: Finding minimal reducts using genetic algorithm. In: Proc. 2nd Annual Join Conf. on Information Sciences, Wrightsville Beach, NC, pp. 186–189 (1995)
25. Zielosko, B., Moshkov, M., Chikalov, I.: Optimization of decision rules based on methods of dynamic programming. Vestnik of Lobachevsky State University of Nizhni Novgorod 6, 195–200 (2010) (in Russian)

Chapter 7
Approaches for Updating Approximations in Set-Valued Information Systems While Objects and Attributes Vary with Time

Hongmei Chen, Tianrui Li, and Hongmei Tian

Abstract. Rough set theory is an important tool for knowledge discovery. The lower and upper approximations are basic operators in rough set theory. Certain and uncertain if-then rules can be unrevealed from different regions partitioned by approximations. In real-life applications, data in the information system are changing frequently, *for example*, objects, attributes, and attributes' values in the information system may vary with time. Therefore, approximations may change over time. Updating approximations efficiently is crucial to the knowledge discovery. The set-valued information system is a general model of the information system. In this chapter, we focus on studying principles for incrementally updating approximations in a set-valued information system while attributes and objects are added. Then, methods for updating approximations of a concept in a set-valued information system is given while attributes and objects change simultaneously. Finally, an extensive experimental evaluation verifies the effectiveness of the proposed method.

Keywords: Knowledge discovery, rough set theory, set-valued information system, approximations.

7.1 Introduction

The Rough Set Theory (RST) proposed by Pawlak is a mathematical tool to process information [18]. It has been successfully applied to knowledge discovery, image

Hongmei Chen · Tianrui Li
School of Information Science and Technology, Southwest Jiaotong University & Key Lab of Cloud Computing and Intelligent Technology, Sichuan Province, Chengdu 610031, China
e-mail: {hmchen,trli}@swjtu.edu.cn

Hongmei Tian
School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
e-mail: 470777848@qq.com

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 229–248.
springerlink.com

processing, and pattern recognition [19, 1, 13, 20, 2, 17]. In Traditional Rough Set theory (TRS), the relation between objects is the equivalence relation. Equivalence classes are induced by different equivalence relations that form a partition of the universe. For any subset of the universe which usually denotes a concept in the information system, equivalence classes are used to describe the concept approximately. If the subset equals to a union of some equivalence classes, then the set is certain. If it cannot be described by certain equivalence classes, then the set is uncertain (rough set). Therefore, a pair of certain sets, *that is*, upper and lower approximations are used to describe the concept approximately. The approximations partition the universe into three different regions. Then, we can induce certain rules from positive region and negative region and uncertain rules from boundary region, *that is*, a tree-way decision [27]. In TRS, the data in the information system are complete, static, and single valued. In real-life applications, attributes' values may be single valued or multi-valued. The set-valued information system is a general model of single-valued information system [11]. For example, the language one can speak may be one of "English, French, German, Chinese, and Japanese" or a power set of multi-language. Set-valued based rough sets can be used to approximate the concept and induce rules in the case of multi-value of attributes. Because of the restriction and uncertain factors to the data access, data missing exists usually. The equivalence relation is substituted by the tolerance relation, the similarity relation, and the limited tolerance relation to process data in real-life applications [14, 24, 26]. Since the value of the missing data may be any values in the attribute values' domain, the set-valued information system has been used to process data in the incomplete information system. In TRS, data in information systems are static, but in real-life applications, information systems are under dynamic environments, *for example*, in the case of processing interactive data and stream data, the limitation of store space, error revision, and addition of data. Objects, attributes, and attributes' values in the information system may vary with time. The approximations of a concept may alter with the variation of the objects, attributes, or attributes' values. How to update approximation effectively is important to the efficiency of the knowledge discovery based on RST.

In [11], Guan defined the tolerance relation, maximal tolerance classes, and relative attribute reduction methods based on maximal tolerance classes in the set-valued information system. Qian et al. defined disjunctive and conjunctive set-valued information systems in [21] in view of different meanings of attributes' values, and they further studied the approach of attribute reduction and rules abstraction in these two different set-valued information systems. Song and Zhang defined partly-accordant reduction and assignment reduction in an inconsistent set-valued decision information system [23]. In [5], Chen et al. introduced the probability rough sets into set-valued information systems, and they studied the method for updating approximations incrementally under variable precision set-valued information system while attributes' values were coarsening and refining.

The attributes, objects and attributes' values may vary when the information system changes which causes the variation of knowledge. Lot of works have been done to study how to update knowledge efficiently when the information system varies

with time [7, 8, 9, 12, 25]. Li et al. proposed the method to update approximations incrementally under the characteristic relation when multi-attributes are added or deleted simultaneously [16]. Cheng proposed two incremental methods for fast computing the approximations in rough fuzzy sets when adding or deleting multi-attributes, *that is*, one starts from the boundary set and the other is based on the cut sets of a fuzzy set [6]. Fan et al. analyzed the different cases of the effect to the rules set when adding objects and they proposed an algorithm to renew rules by updating Strength Index (SI) [10]. Zheng and Wang proposed a rule tree, and they investigated the methods to renew the rule set by updating the rule tree [28]. Skowron et al. defined the function approximations that are used to induce unknown rules through known object sets [22]. Liu et al. defined the covering matrix and precision matrix in [15]. Then, they proposed methods to renew interesting knowledge by updating covering matrix and precision matrix under the immigrant and emigrant of objects. Chen et al. studied the method to update approximations incrementally by analyzing the change of granularity and boundary region while attributes' values were coarsening and refining [3, 29]. When attributes and objects change simultaneously, Chen et al. proposed the method to update approximations in TRS [4]. In the dynamic data environment, attributes and object may vary simultaneously. In this case, there is no literature on how to update knowledge in extended rough sets. In this chapter, we investigate how to update approximations by decomposing and combining the information system and using the information in the boundary region. We also investigate how to update approximations by an accumulation policy. The work will improve the efficiency of knowledge discovery and further promote the study of the updating method for knowledge discovery in the condition of the complex change of data.

The chapter is organized as follows. In Section 7.2, we introduce the basic concepts in the set-valued rough set theory, such as the dominance relation in the set-valued information system and the definitions of upper and lower approximations. In Section 7.3, we study methods for updating approximations incrementally while adding attributes and objects adding simultaneously. In Section 7.4, an accumulation method to update approximations incrementally is proposed. In Section 7.5, an illustrative example is given to show the decomposing method used to update approximations. In Section 7.6, extensive experiments have been carried out to verify the efficiency of the method. We conclude the chapter in the Section 7.7 and outline the direction of our further work.

7.2 Rough Set Theory in a Set-Valued Information System

In this section, we recall basic concepts in a set-valued information system.

Definition 7.2.1. *Let* $S = (U, A, V, f)$ *be a set-valued information system, where* $U = \{x_1, x_2, \ldots, x_n\}$ *is a non-empty finite set of objects, called the universe.* $A = \{a_1, a_2, \ldots, a_l\}$ *is a non-empty finite set of attributes. The element in* A *is called an attribute.* $A = C \cup D$, $C \cap D = \emptyset$, C *is the set of condition attributes, and* D *is the*

set of decision attributes. $V = \{V_{a_i} | a_i \in A\}$, $V_{a_i}(i = 1,2,\cdots,l)$ *is the domain of attribute* $a_i(a_i \in A)$. V_C *is the domain of condition attributes.* V_D *is the domain of decision attributes.* $V = V_C \cup V_D$ *is the domain of all attributes.* $f : U \times C \to 2^{V_C}$ *is a set-valued mapping.* $f : U \times D \to V_D$ *is a single-valued mapping.*

Definition 7.2.2. *Given a set-valued information system* $S = (U,A,V,f)$, *for* $B \subseteq C$, *a set-valued dominance relation is defined as follows:*

$$R_B^{\geq} = \{(y,x) \in U \times U \,|\, f(y,a) \supseteq f(x,a), \forall a \in B\} \tag{7.1}$$

$R_B^{\geq}$ is reflexive, asymmetric, and transitive. We denote $[x]_B^{\geq} = \{y \in U \,|\, (y,x) \in R_B^{\geq}\}$, $[x]_B^{\leq} = \{y \in U \,\big|\, (x,y) \in R_B^{\geq}\}$. $[x]_B^{\geq}(x \in U)$ is a granule of knowledge induced by the dominance relation, which are the set of objects dominating x.

Definition 7.2.3. $\forall B \subseteq A$, $\forall X \subseteq U$, *the lower and the upper approximations of* X *under the dominance relation* $R_B^{\geq}$ *are defined respectively as follows.*

$$\underline{R_B^{\geq}}(X) = \{x \in U \,\big|\, [x]_B^{\geq} \subseteq X\} \text{ and } \overline{R_B^{\geq}}(X) = \{x \in U \,\big|\, [x]_B^{\geq} \cap X \neq \emptyset\} \tag{7.2}$$

Based on the approximations of X, one can partition the universe U into three disjoint regions, *that is*, the positive region $POS^{\geq}(X)$, the boundary region $BNR^{\geq}(X)$, and the negative region $NEG^{\geq}(X)$:

$$POS^{\geq}(X) = \underline{R_B^{\geq}}(X); \tag{7.3}$$

$$BNR^{\geq}(X) = \overline{R_B^{\geq}}(X) - \underline{R_B^{\geq}}(X); \tag{7.4}$$

$$NEG^{\geq}(X) = U - \overline{R_B^{\geq}}(X). \tag{7.5}$$

We can obtain certain rules from the position region and negative region and possible rules from the boundary region.

7.3 Principle for Dynamic Maintenance of Approximations While Objects and Attributes Are Added Simultaneously

In this section, we study the principle for dynamic maintenance of approximations in a set-valued information system under the addition of objects and attributes.

Let $S^t = (U^t, A^t, V^t, f^t)$ denote the set-valued information system at time t, where $A^t = C^t \cup d^t$, $|U^t| = n$, $|C^t| = m$. Let U^+ denote a set of objects added to the set-valued information system, and C^+ denote a set of conditional attributes added to the set-valued information system. $|U^+| = n^+$, $U^+ = \{x_j | j = n+1, n+2, \ldots, n+n^+\}$, $|C^+| = m^+$, $C^+ = \{a_j | j = m+1, m+2, \ldots, m+m^+\}$. In this chapter, we only consider the case when objects and conditional attributes are added

to the set-valued information system but decision attributes are kept unchanged. Let $S^{t+1} = (U^{t+1}, A^{t+1}, V^{t+1}, f^{t+1})$ denote the set-valued information system at time $t+1$, where $U^{t+1} = U^t \cup U^+$, $A^{t+1} = C^{t+1} \cup d^t$, $C^{t+1} = C^t \cup C^+$. Then, we decompose the set-valued information system S^{t+1} into two set-valued information systems, *that is*, $S^{U+} = (U^+, A^{t+1}, V^{U+}, f^{U+})$, $S^{A+} = (U^t, A^{t+1}, V^{A+}, f^{A+})$. The set-valued information system S^{A+} is further decomposed into two sub set-valued information systems $S^t = (U^t, A^t, V^t, f^t)$ and $S^{\Delta A} = (U^t, A^{\Delta A}, V^{\Delta}, f^{\Delta})$, where $A^{\Delta A} = C^+ \cup d^t$. Note that the attributes' domain may change when adding objects to the information system. Let $[x_i^S]_B^{\geq}$ denote dominating classes of object x_i induced by B in the set-valued information system S, where $B \subseteq C$.

Now we consider the variation of granules induced by the dominating relation in a set-valued information system.

Lemma 7.3.1. $[x_i^{S^{A+}}]_{C^{t+1}}^{\geq} = [x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq}$.

Proof. By Definition 7.2.2, $[x_i^{S^{A+}}]_{C^{t+1}}^{\geq} = \{y \in U^t \big| f(y,a) \supseteq f(x_i,a), \forall a \in C^{t+1}\}$, $[x_i^{S^t}]_{C^t}^{\geq} = \{y \in U^t \,|\, f(y,a) \supseteq f(x_i,a), \forall a \in C^t\}$, $[x_i^{S^{\Delta A}}]_{C^+}^{\geq} = \{y \in U^t \,|\, f(y,a) \supseteq f(x_i,a), \forall a \in C^+\}$. Since $C^{t+1} = C^t \cup C^+$, then $[x_i^{S^{A+}}]_{C^{t+1}}^{\geq} = [x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq}$.

From Lemma 7.3.1, the granules in the set-valued information system may become smaller with the addition of attributes.

Lemma 7.3.2. *For* $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq}$, *the following hold:*

1. *If* $\exists x_{n+j}^{S^{U+}}$ *s.t.* $x_{n+j}^{S^{U+}} R_{C^{t+1}}^{\geq} x_i^{S^{A+}}$, *then* $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_i^{S^{A^+}}]_{C^{t+1}}^{\geq} \cup [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq}$;
2. *If* $\neg\exists x_{n+j}^{S^{U+}}$ *s.t.* $x_{n+j}^{S^{U+}} R_{C^{t+1}}^{\geq} x_i^{S^{A+}}$, *then* $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_i^{S^{A^+}}]_{C^{t+1}}^{\geq}$;
3. *If* $\exists x_i^{S^{A+}}$ *s.t.* $x_i^{S^{A+}} R_{C^{t+1}}^{\geq} x_{n+j}^{S^{U+}}$, *then* $[x_{n+j}^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq} \cup [x_i^{S^{A^+}}]_{C^{t+1}}^{\geq}$;
4. *If* $\neg\exists x_i^{S^{A+}}$ *s.t.* $x_i^{S^{A+}} R_{C^{t+1}}^{\geq} x_{n+j}^{S^{U+}}$, *then* $[x_{n+j}^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq}$.

Proof. 1. By Definition 7.2.2, if $\exists x_{n+l}^{S^{U+}} \in [x_{n+j}^{S^{U^+}}]_{C^{t+1}}^{\geq}$, then $x_{n+l}^{S^{U+}} R_{C^{t+1}}^{\geq} x_{n+j}^{S^{U+}}$. Because the transitive of the dominance relation, $x_{n+j}^{S^{U+}} R_{C^{t+1}}^{\geq} x_i^{S^{A+}}$. Thus $x_{n+l}^{S^{U+}} R_{C^{t+1}}^{\geq} x_i^{S^{A+}}$. That is, $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_i^{S^{A^+}}]_{C^{t+1}}^{\geq} \cup [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq}$.

Analogously, 2., 3. and 4. hold.

Lemma 7.3.2 explains the four kinds of change of a granule with respect to the addition of objects in a set-valued information system.

Let $C_{i1} = \{x_{n+j}^{S^{U+}} \big| x_{n+j}^{S^{U+}} R_{C^{t+1}}^{\geq} x_i^{S^{A+}}\}$, $C_{j2} = \{x_i^{S^{A+}} \big| x_i^{S^{A+}} R_{C^{t+1}}^{\geq} x_{n+j}^{S^{U+}}\} (1 \leq i \leq n,\ 1 \leq j \leq n^+)$. From Lemmas 7.3.1 and 7.3.2, the granules of a set-valued information system will be one of the following cases when attributes and objects are added simultaneously.

i. If $C_{i1} \neq \emptyset$, then $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq} = ([x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq}) \cup C_{i1}$;

ii. If $C_{i1} = \emptyset$, then $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq}$;

iii. If $C_{j2} \neq \emptyset$, then $[x_{n+j}^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_{n+j}^{S^{U^+}}]_{C^{t+1}}^{\geq} \cup C_{j2}$;

iv. If $C_{j2} = \emptyset$, then $[x_{n+j}^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq}$.

Let $\underline{R_{B,S}^{\geq}}(X)$, $\overline{R_{B,S}^{\geq}}(X)$ denote lower and upper approximations of a concept X in the set-valued information system S, respectively. For example, $\underline{R_{C^t,S^t}^{\geq}}(X^t)$, $\overline{R_{C^t,S^t}^{\geq}}(X^t)$ denote the lower and upper approximations of the concept X^t in the set-valued information system S^t, respectively. The cardinality of a concept in the set-valued information system S^t may vary when adding objects to the set-valued information system, *that is*, $X^{t+1} = X^t \cup X^{U^+}$. Without loss of generality, $X^{t+1} = \{x_i | f(x_i, d) = k, 1 \leq i \leq n+n^+\}$. The concept X is approximated by dominating classes in a set-valued information system. The approximations may change with the variation of the granules and the concept. The following propositions hold.

Proposition 7.3.1. *If* $[x_i^{S^{t+1}}]_{C^{t+1}}^{\geq} = [x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \cup C_{i1}$ *($C_{i1} \neq \emptyset$), then the following hold:*

i. *If* $x_i^{S^t} \in BNR_{C^t,S^t}^{\geq}(X^t)$, $x_i^{S^t} \in X^t$, $[x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \subseteq X^t$, $\forall x_{n+j}^{S^{U+}} \in C_{i1}$, $x_{n+j}^{S^{U+}} \in \underline{R_{C^{t+1},S^{U^+}}^{\geq}}(X^{U^+})$, *then* $\underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \underline{R_{C^t,S^t}^{\geq}}(X^t) \cup \{x_i^{S^{t+1}}\}$, $\overline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \overline{\overline{R_{C^t,S^t}^{\geq}}(X^t)}$;

ii. *If* $x_i^{S^t} \in \underline{R_{C^t,S^t}^{\geq}}(X^t)$, $\exists x_{n+j}^{S^{U+}} \in C_{i1}$, $x_{n+j}^{S^{U+}} \notin \underline{R_{C^{t+1},S^{U^+}}^{\geq}}(X^{U^+})$, *then* $x_i^{S^{t+1}} \notin \underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1})$, $\underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \underline{R_{C^t,S^t}^{\geq}}(X^t) - \{x_i^{S^t}\}$, $\overline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \overline{R_{C^t,S^t}^{\geq}}(X^t)$;

iii. *If* $x_i^{S^t} \notin \overline{R_{C^t,S^t}^{\geq}}(X^t)$, $\exists x_{n+j}^{S^{U+}} \in C_{i1}$, $x_{n+j}^{S^{U+}} \in \underline{R_{C^{t+1},S^{U^+}}^{\geq}}(X^{U^+})$, *then* $\underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \underline{R_{C^t,S^t}^{\geq}}(X^t)$, $\overline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \overline{R_{C^t,S^t}^{\geq}}(X^t) \cup \{x_i^{S^{t+1}}\}$;

iv. *Otherwise,* $\underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \underline{R_{C^t,S^t}^{\geq}}(X^t)$, $\overline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \overline{R_{C^t,S^t}^{\geq}}(X^t)$.

Proof. i. If $x_i^{S^t} \in BNR_{C^t,S^t}^{\geq}(X^t)$, $[x_i^{S^t}]_{C^t}^{\geq} \cap X^t \neq \emptyset$, $[x_i^{S^t}]_{C^t}^{\geq} \not\subset X^t$. If $x_i^{S^t} \notin X^t$, then $x_i^{S^t} \in [x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq}$, $[x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \not\subset X^t$. If $x_i^{S^t} \in X^t$, then $[x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \subseteq X^t$, $x_i^{S^{A+}} \in \underline{R_{C^{t+1},S^{A^+}}^{\geq}}(X^t)$, $[x_i^{S^{A+}}]_{C^{t+1}}^{\geq} \subseteq X^t$. Since $\forall x_{n+j}^{S^{U+}} \in C_{i1}$, $x_{n+j}^{S^{U+}} \in \underline{R_{C^{t+1},S^{U^+}}^{\geq}}(X^{U^+})$, then $x_{n+j}^{S^{U+}} \subseteq X^{U^+}$. Because $X^{t+1} = X^t \cup X^{U+}$, $[x_i^{S^{A+}}]_{C^{t+1}}^{\geq} \cup \{x_{n+j}^{S^{U+}}\} \subseteq X^t \cup X^{U^+}$, that is, $([x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq}) \cup C_{i1} \subseteq X^t \cup X^{U^+}$. Since $x_i^{S^t} \notin \underline{R_{C^t,S^t}^{\geq}}(X^t)$, $x_i^{S^t} \in BNR_{C^t,S^t}^{\geq}(X^t)$, $x_i^{S^t} \in \overline{R_{C^t,S^t}^{\geq}}(X^t)$, and $x_i^{S^{t+1}} \in \underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1})$. Therefore, $\underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \underline{R_{C^t,S^t}^{\geq}}(X^t) \cup \{x_i^{S^{t+1}}\}$, $\overline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \overline{R_{C^t,S^t}^{\geq}}(X^t)$.

ii. If $x_i^{S^t} \in \underline{R_{C^t,S^t}^{\geq}}(X^t)$, $\exists x_{n+j}^{S^{U+}} \in C_{i1}$, $x_{n+j}^{S^{U+}} \notin \underline{R_{C^{t+1},S^{U^+}}^{\geq}}(X^{U^+})$, then $[x_i^{S^t}]_{C^t}^{\geq} \subseteq X^t$, $[x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq} \not\subset X^{U^+}$. Thus $[x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \subseteq X^t$, $[x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \cup [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq} \not\subset X^t \cup X^{U^+}$, $([x_i^{S^t}]_{C^t}^{\geq} \cap [x_i^{S^{\Delta A}}]_{C^+}^{\geq} \cup [x_{n+j}^{S^{U+}}]_{C^{t+1}}^{\geq}) \cap (X^t \cup X^{U^+}) \neq \emptyset$. Therefore, $\underline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \underline{R_{C^t,S^t}^{\geq}}(X^t) - \{x_i^{S^t}\}$, $\overline{R_{C^{t+1},S^{t+1}}^{\geq}}(X^{t+1}) = \overline{R_{C^t,S^t}^{\geq}}(X^t)$.

iii. If $x_i^{S^t} \notin \overline{R^{\geq}_{C^t,S^t}}(X^t)$, $\exists x_{n+j}^{S^{U+}} \in C_{i1}$, $x_{n+j}^{S^{U+}} \in \underline{R^{\geq}_{C^{t+1},S^{U^+}}}(X^{U^+})$, then $[x_i^{S^t}]^{\geq}_{C^t} \cap X^t = \emptyset$, $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \subseteq X^{U^+}$. Thus $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}) \cap X^t = \emptyset$, $[x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+} \cup [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \not\subset X^t \cup X^{U^+}$, $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+} \cup [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}}) \cap (X^t \cup X^{U^+}) \neq \emptyset$. Therefore, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_i^{S^{t+1}}\}$.

iv. The proof is similar to the proof of i, ii.

Proposition 7.3.2. *If* $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}$, *then the following hold:*

i. If $x_i^{S^t} \in BNR^{\geq}_{C^t,S^t}(X)$,

a) If $x_i^{S^t} \in X^t$, $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}) \subseteq X^t$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_i^{S^{t+1}}\}$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$;

b) If $x_i^{S^t} \notin X^t$, $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}) \cap X^t = \emptyset$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) - \{x_i^{S^{t+1}}\}$;

ii. Otherwise, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$.

Proof. i. a) If $x_i^{S^t} \in X^t$, $[x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+} \subseteq X^t$, then $[x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+} \subseteq (X^t \cup X^{U^+})$. $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} \subseteq X^{t+1}$, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_i^{S^{t+1}}\}$. Since $x_i^{S^t} \in BNR^{\geq}_{C^t,S^t}(X)$, then $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$. b) If $x_i^{S^t} \notin X^t$, $[x_i^{S^t}]^{\geq}_{C^t} \not\subset X^t$, then if $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}) \cap X^t = \emptyset$, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) - \{x_i^{S^{t+1}}\}$.

ii. Because $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}$, $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} \subseteq ([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+})$. If $[x_i^{S^t}]^{\geq}_{C^t} \subseteq X^t$, then $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} \subseteq X^t$. If $[x_i^{S^t}]^{\geq}_{C^t} \cap X^t = \emptyset$, then $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} \cap X^t = \emptyset$. That is, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$.

Proposition 7.3.3. *If* $[x_{n+j}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{n+j}^{S^{U^+}}]^{\geq}_{C^{t+1}} \cup C_{j2} (C_{j2} \neq \emptyset)$, *then the following hold:*

i. If $x_{n+j}^{S^{U+}} \in \underline{R^{\geq}_{C^{t+1},S^{U^+}}}(X^{U^+})$, $\forall x_i^{S^{A+}} \in C_{j2}$, $f(x_i^{S^{A+}}, d) = k$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$;

ii. If $x_{n+j}^{S^{U+}} \in BNR^{\geq}_{C^{t+1},S^{U^+}}(X^{U^+})$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$;

iii. If $x_{n+j}^{S^{U+}} \notin \overline{R^{\geq}_{C^{t+1},S^{U^+}}}(X^{U^+})$, $\exists x_i^{S^{A+}} \in C_{j2}$, $f(x_i^{S^{A+}}, d) = k$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$.

iv. Otherwise, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$.

Proof. i. If $x_{n+j}^{S^{U+}} \in \underline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, then $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \subseteq X^{U+}$. Because $\forall x_i^{S^{A+}} \in C_{j2}$, $f(x_i^{S^{A+}},d)=k$, then if $C_{j2} \subseteq X^t$, $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cup C_{j2} \subseteq (X^t \cup X^{U+})$. That is, $[\underline{x_{n+j}^{S^{t+1}}}]^{\geq}_{C^{t+1}} \subseteq X^{t+1}$. Then $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$.

ii. If $x_{n+j}^{S^{U+}} \in BNR^{\geq}_{C^{t+1},S^{U+}}(X^{U+})$, then $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \not\subset X^{U+}$, $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cap X^{U+} \neq \emptyset$. Thus $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cup C_{j2} \not\subset (X^t \cup X^{U+})$, $([x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cup C_{j2}) \cap (X^t \cup X^{U+}) \neq \emptyset$, that is, $x_{n+j}^{S^{U+}} \in BNR^{\geq}_{C^{t+1},S^{t+1}}(X^{t+1})$. Therefore, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$.

iii. If $x_{n+j}^{S^{U+}} \notin \overline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, then $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cap X^{U+} = \emptyset$. If $\exists x_i^{S^{A+}} \in C_{j2}$, $f(x_i^{S^{A+}},d) = k$, then $C_{j2} \cap X^t \neq \emptyset$, $([x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cup C_2) \not\subset (X^t \cup X^{U+})$, $([x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cup C_{j2}) \cap (X^t \cup X^{U+}) \neq \emptyset$. Therefore, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$.

iv. If $x_{n+j}^{S^{U+}} \notin \overline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, $C_{j2} \cap X^t = \emptyset$, then $([x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cup C_{j2}) \cap (X^t \cup X^{U+}) = \emptyset$. Therefore, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$.

Proposition 7.3.4. *If* $[x_{n+j}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}}$, *then the following hold:*

i. If $x_{n+j}^{S^{U+}} \in \underline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$;*

ii. If $x_{n+j}^{S^{U+}} \in BNR^{\geq}_{C^{t+1},S^{U+}}(X^{U+})$, *then* $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$;*

iii. Otherwise, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$.

Proof. i. If $x_{n+j}^{S^{U+}} \in \underline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, then $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \subseteq X^{U+}$. Therefore, $[x_{n+j}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \subseteq (X^t \cup X^{U+})$, that is, $x_{n+j}^{S^{t+1}} \in \underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$. Thus $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$.

ii. If $x_{n+j}^{S^{U+}} \in BNR^{\geq}_{C^{t+1},S^{U+}}(X^{U+})$, then $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \not\subset X^{U+}$, $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cap X^{U+} \neq \emptyset$. Therefore, $[x_{n+j}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \not\subset (X^t \cup X^{U+})$, $[\underline{x_{n+j}^{S^{t+1}}}]^{\geq}_{C^{t+1}} = [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cap (X^t \cup X^{U+}) \neq \emptyset$, that is, $x_{n+j}^{S^{t+1}} \notin \underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $x_{n+j}^{S^{t+1}} \in \overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$. Therefore, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$.

iii. If $x_{n+j}^{S^{U+}} \notin \overline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, then $[x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cap X^{U+} = \emptyset$. Therefore, $[x_{n+j}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{n+j}^{S^{U+}}]^{\geq}_{C^{t+1}} \cap (X^t \cup X^{U+}) \neq \emptyset$. Thus $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t)$.

Considering the variation of granules in a set-valued information system with respect to attributes and objects are added simultaneously, we discuss the change of lower and upper approximations in Propositions 7.3.1, 7.3.2, 7.3.3 and 7.3.4. In the following, we present algorithms (see Algorithms 7.1, 7.2 and 7.3) for updating approximations of a concept when objects and attributes are added simultaneously.

Algorithm 7.1. Algorithm for Updating Approximations when Objects and Attributes are Added based on the Decomposition (AUAOAAD)

INPUT: $\underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^t,S^t}}(X^t)$, $[x_i]^{\geq}_{C^t}$ $(1 \leq i \leq |U^t|)$, X^t, U^+, A^+

OUTPUT: $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

for $i = 1$ to n **do**
  Compute $[x_i^{S^{\Delta A}}]^{\geq}_{C^+}$ in $S^{\Delta A}$
  $[x_i^{S^{A+}}]^{\geq}_{C^{t+1}} \leftarrow [x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}$;
end for
Compute $[x_j^{S^{U+}}]^{\geq}_{C^+}$, $\underline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$, $\overline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$
for $i = 1$ to n **do**
  $sig_i = 0$, $sign_i$=0, $C_{i1} = 0$, $C_{j2} = 0$
  for $j = n+1$ to $n+n^+$ **do**
    if $x_j^{S^{U+}} R^{\geq}_{C^{t+1}} x_i^{S^{A+}}$ **then**
      $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}} \leftarrow [x_i^{S^{A+}}]^{\geq}_{C^{t+1}} \cup \{x_j^{S^{U+}}\}$, $C_{i1} \leftarrow 1$
      if $x_j^{S^{U+}} \in \underline{R^{\geq}_{C^{t+1},S^{U+}}}(X^{U+})$ **then**
        $sig_i \leftarrow 1$
      else
        $\text{sign}_i \leftarrow 1$;
      end if
    else if $x_i^{S^{A+}} R^{\geq}_{C^{t+1}} x_j^{S^{U+}}$ **then**
      $[x_j^{S^{t+1}}]^{\geq}_{C^{t+1}} \leftarrow [x_j^{S^{U+}}]^{\geq}_{C^{t+1}} \cup \{x_i^{S^{A+}}\}$, $C_{j2} \leftarrow 1$
    end if
  end for
  AUADCV($\underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^t,S^t}}(X^t)$, C_{i1}, $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}}$)
end for
for $j = n+1$ to $n+n^+$ **do**
  AUAUDC($\underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^t,S^t}}(X^t)$,$C_{j2}$, $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}}$)
end for
return $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

7.4 Accumulation Principle for Dynamic Maintenance of Approximations While Objects and Attributes Are Added

When objects and attributes are added simultaneously, we may see it as two sequential courses, *that is*, objects are added firstly and attributes are added secondly (or in a reverse order). When objects (attributes) are added to the set-valued information system, it may be seen as the accumulation results of objects (attributes) are added to the set-valued information system one by one. Then, if we develop an effective method for updating approximations when a single object (attribute) is added, then we can deal with the case when multi-objects and multi-attributes are added to the information system simultaneously. We present the principle for updating approximations when a single attribute or object is added to the set-valued information system as follows.

Algorithm 7.2. Algorithm for Updating Approximations when Dominating Classes in S^{A+} Vary (AUADCV)

INPUT: $\underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^t,S^t}}(X^t)$, $[x_i^{S^{t+1}}]^{\geq}_{C^{t+1}}$ $(1 \leq i \leq |U^t|)$, X^t

OUTPUT: $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

if $C_{i1} = 1$ **then**
  if $x_i^{S^t} \in BNR^{\geq}_{C^t,S^t}(X^t), x_i^{S^t} \in X^t, [x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+} \subseteq X^t, sig_i = 1, sign_i = 0$ **then**
    $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_i^{S^{t+1}}\}$
  else if $x_i^{S^t} \in \underline{R^{\geq}_{C^t,S^t}}(X^t), sign_i = 1$ **then**
    $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \underline{R^{\geq}_{C^t,S^t}}(X^t) - \{x_i^{S^t}\}$
  else if $x_i^{S^t} \notin \overline{R^{\geq}_{C^t,S^t}}(X^t), sig_i = 1$ **then**
    $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_i^{S^{t+1}}\}$
  end if
else
  if $x_i^{S^t} \in BNR^{\geq}_{C^t,S^t}(X)$ **then**
    if $x_i^{S^t} \in X^t$, $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}) \subseteq X^t$ **then**
      $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_i^{S^{t+1}}\}$;
    else if $x_i^{S^t} \notin X^t$, $([x_i^{S^t}]^{\geq}_{C^t} \cap [x_i^{S^{\Delta A}}]^{\geq}_{C^+}) \cap X^t = \emptyset$ **then**
      $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\geq}_{C^t,S^t}}(X^t) - \{x_i^{S^{t+1}}\}$;
    end if
  end if
end if
return $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

Let $S^t = (U^t, A^t)$ be a set-valued information system, and $S^{t'} = (U^{t'}, A^{t'})$ denotes the set-valued information system after adding an object into the set-valued information system, where $U^{t'} = U^t \cup \{x_{n+1}\}(|U^t| = n), A^{t'} = A^t, X^t = \{x_i \,|\, f(x_i, d) = k, 1 \leq$

$i \le n\}$. $[x_i]_B^{\succeq t'}$ denotes dominating class of object x_i induced by B in the set-valued information system $S^{t'}$, where $B \subseteq C$. Then, we give the principle for updating approximations when an object is added.

Algorithm 7.3. Algorithm for Updating Approximations when Updating Dominating Classes in S^{U^+} Vary (AUAUDC)

INPUT: $\underline{R^{\succeq}_{C^t,S^t}}(X^t)$, $\overline{R^{\succeq}_{C^t,S^t}}(X^t)$, C_{j2}, $[x_i^{S^{t+1}}]^{\succeq}_{C^{t+1}}$

OUTPUT: $\underline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $\overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

1: **if** $C_{j2} = 1$ **then**
2: **if** $x_{n+j}^{S^{U^+}} \in \underline{R^{\succeq}_{C^{t+1},S^{U^+}}}(X^{U^+}), \forall x_i^{S^{A+}} \in C_2, f(x_i^{S^{A+}}, d) = k$ **then**
3: $\underline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \underline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}, \overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$
4: **else if** $x_{n+j}^{S^{U^+}} \in BNR^{\succeq}_{C^{t+1},S^{U^+}}(X^{U^+})$ **then**
5: $\overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$
6: **else if** $x_{n+j}^{S^{U^+}} \notin \overline{R^{\succeq}_{C^{t+1},S^{U^+}}}(X^{U^+}), \exists x_i^{S^{A+}} \in C_2, f(x_i^{S^{A+}}, d) = k$ **then**
7: $\overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$
8: **end if**
9: **else**
10: **if** $x_{n+j}^{S^{U^+}} \in \underline{R^{\succeq}_{C^{t+1},S^{U^+}}}(X^{U^+})$ **then**
11: $\underline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \underline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}, \overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$
12: **else if** $x_{n+j}^{S^{U^+}} \in BNR^{\succeq}_{C^{t+1},S^{U^+}}(X^{U^+})$ **then**
13: $\overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}) \leftarrow \overline{R^{\succeq}_{C^t,S^t}}(X^t) \cup \{x_{n+j}^{S^{t+1}}\}$;
14: **end if**
15: **end if**
16: **return** $\underline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1}), \overline{R^{\succeq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

Lemma 7.4.1. *For* $[x_i]_B^{\succeq t'} (1 \le i \le n)$, *if* $(x_{n+1}, x_i) \in R_B^{\succeq}$, *then* $[x_i]_B^{\succeq t'} = [x_i]_B^{\succeq t} \cup \{x_{n+1}\}$.

Proof. It is directly from Definition 7.2.2, if $f(x_{n+1}, d) = k$, then $X^{t'} = X^t \cup \{x_{n+1}\}$; If $f(x_{n+1}, d) \ne k$, then $X^{t'} = X^t$. The following proposition holds when an object is added to the set-valued information system.

Proposition 7.4.1. *For* $\underline{R_B^{\succeq}}(X^{t'})$, $\overline{R_B^{\succeq}}(X^{t'})$, *the following hold:*

i. If $f(x_{n+1}, d) = k$, $[x_i]_B^{\succeq t'} \ne [x_i]_B^{\succeq t}$, *and* $x_i \notin \overline{R_B^{\succeq}}(X^t)$, *then* $\underline{R_B^{\succeq}}(X^{t'}) = \underline{R_B^{\succeq}}(X^t)$, $\overline{R_B^{\succeq}}(X^{t'}) = \overline{R_B^{\succeq}}(X^t) \cup \{x_i\}$;
ii. If $f(x_{n+1}, d) \ne k$, $[x_i]_B^{\succeq t'} \ne [x_i]_B^{\succeq t}$, *and* $x_i \in \underline{R_B^{\succeq}}(X^t)$, *then* $\underline{R_B^{\succeq}}(X^{t'}) = \underline{R_B^{\succeq}}(X^t) - \{x_i\}$, $\overline{R_B^{\succeq}}(X^{t'}) = \overline{R_B^{\succeq}}(X^t)$;

iii. If $[x_{n+1}]_B^{\geq t'} \cap X^{t'} \neq \emptyset$, *then* $\underline{R_B^{\geq}}(X^{t'}) = \underline{R_B^{\geq}}(X^t)$, $\overline{R_B^{\geq}}(X^{t'}) = \overline{R_B^{\geq}}(X^t) \cup \{x_{n+1}\}$;

iv. If $[x_{n+1}]_B^{\geq t'} \subseteq X^{t'}$, *then* $\underline{R_B^{\geq}}(X^{t'}) = \underline{R_B^{\geq}}(X^t) \cup \{x_{n+1}\}$, $\overline{R_B^{\geq}}(X^{t'}) = \overline{R_B^{\geq}}(X^t) \cup \{x_{n+1}\}$;

Proof. i. If $f(x_{n+1}, d) = k$, then $X^{t'} = X^t \cup \{x_{n+1}\}$. Because $x_i \notin \overline{R_B^{\geq}}(X^t)$, then $[x_i]_B^{\geq t} \cap X^t = \emptyset$. Therefore, $[x_i]_B^{\geq t'} = ([x_i]_B^{\geq t} \cup \{x_{n+1}\}) \cap X^{t'} \neq \emptyset$. Thus, $\underline{R_B^{\geq}}(X^{t'}) = \underline{R_B^{\geq}}(X^t)$, $\overline{R_B^{\geq}}(X^{t'}) = \overline{R_B^{\geq}}(X^t) \cup \{x_i\}$.

ii. If $f(x_{n+1}, d) = k$, then $X^{t'} = X^t$. Since $x_i \in \underline{R_B^{\geq}}(X^t)$, then $[x_i]_B^{\geq t} \subseteq X^t$. Because $[x_i]_B^{\geq t'} = ([x_i]_B^{\geq t} \cup \{x_{n+1}\}) \not\subset X^{t'}$, then $\underline{R_B^{\geq}}(X^{t'}) = \underline{R_B^{\geq}}(X^t) - \{x_i\}$, $\overline{R_B^{\geq}}(X^{t'}) = \overline{R_B^{\geq}}(X^t)$. The proofs of iii and iv are directly from Definition 7.2.3.

Let $S^t = (U^t, A^t)$ be a set-valued information system, and $S^{t''} = (U^{t''}, A^{t''})$ denote the set-valued information system after adding an attribute into the set-valued information system, where $U^{t''} = U^t (|U^t| = n)$, $A^{t''} = A^t \cup \{a_{m+1}\} (|A^t| = m)$, $X^t = \{x_i \,|\, f(x_i, d) = k, 1 \leq i \leq n\}$. $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''}$ denotes dominating class of object x_i induced by $B \cup \{a_{m+1}\}$ in the set-valued information system $S^{t''}$. The following proposition holds when an attribute is added to the set-valued information system.

Lemma 7.4.2. $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''} = [x_i]_B^{\geq t} \cap [x_i]_{\{a_{m+1}\}}^{\geq t} (1 \leq i \leq n)$.

Proof. It follows directly from Definition 7.2.2.

Proposition 7.4.2. *If* $f(x_i, d) = k$, $x_i \in BNR^{\geq}(X^t)$, $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''} \subseteq X^t$, *then* $\underline{R_B^{\geq}}(X^{t''}) = \underline{R_B^{\geq}}(X^t) \cup \{x_i\}$.

Proof. From Lemma 7.4.2, $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''} \subseteq [x_i]_B^{\geq t}$. Then if $f(x_i, d) \neq k$, $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''} \not\subset X^t$ due to the reflexivity of $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''}$. Therefore, if $f(x_i, d) = k$, $x_i \in BNR^{\geq}(X^t)$, $[x_i]_{B \cup \{a_{m+1}\}}^{\geq t''} \subseteq X^t$, then $\underline{R_B^{\geq}}(X^{t''}) = \underline{R_B^{\geq}}(X^t) \cup \{x_i\}$ by Definition 7.2.2.

Let $S^{t+1} = (U^{t+1}, A^{t+1})$ denote the set-valued information system after adding attributes into the set-valued information system, where $U^{t+1} = U^t \cup U^+ (|U^t| = n, |U^+| = n^+)$, $A^{t+1} = A^t \cup A^+ (|A^t| = m, |A^+| = m^+)$, $X^t = \{x_i \,|\, f(x_i, d) = k, 1 \leq i \leq n\}$. In the following, we propose an accumulation algorithm (see Algorithm 7.4) for updating approximations while objects and attributes are added simultaneously.

7.5 Example

In this section, an example is used to explain the methods proposed in this chapter. The computing procedure of Algorithm 7.1 is also explained by the example as follows.

Algorithm 7.4. Accumulation Algorithm for Updating Approximations when Objects and Attributes are Added (AAUAOAA)

INPUT $\underline{R^{\geq}_{C^t,S^t}}(X^t)$, $\overline{R^{\geq}_{C^t,S^t}}(X^t)$, $[x_i]^{\geq}_{C^t}$ $(1 \leq i \leq |U^t|)$, X^t, U^+, A^+

OUTPUT $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$,$\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

for $i = 1$ to n **do**
  for $j = n+1$ To $n+n^+$ **do**
    $[x_j]^{\geq t'}_{C^t} \leftarrow \{x_j\}$
    if $x_j R^{\geq}_B x_i$ **then**
      $[x_i]^{\geq t'}_{C^t} \leftarrow [x_i]^{\geq t}_{C^t} \cup \{x_j\}$
    end if
    if $f(x_j,d) = k, x_i \notin \overline{R^{\geq}_B}(X^t))$ **then**
      $\overline{R^{\geq}_B}(X^{t'}) = \overline{R^{\geq}_B}(X^t) \cup \{x_i\}$
    else if $f(x_j,d) \neq k, x_i \in \underline{R^{\geq}_B}(X^t))$ **then**
      $\underline{R^{\geq}_B}(X^{t'}) = \underline{R^{\geq}_B}(X^t) - \{x_i\}$
    end if
    if $x_i R^{\geq}_B x_j$ **then**
      $[x_j]^{\geq t'}_{C^t} \leftarrow [x_j]^{\geq t}_{C^t} \cup \{x_i\}$
    end if
  end for
end for
for $j = n+1$ To $n+n^+$ **do**
  if $[x_j]^{\geq t'}_{C^t} \cap X^{t'} \neq \emptyset$ **then**
    $\overline{R^{\geq}_B}(X^{t'}) = \overline{R^{\geq}_B}(X^t) \cup \{x_j\}$
  end if
  if $[x_j]^{\geq t'}_{C^t} \subseteq X^{t+1}$ **then**
    $\underline{R^{\geq}_B}(X^{t'}) = \underline{R^{\geq}_B}(X^t) \cup \{x_j\}$, $\overline{R^{\geq}_B}(X^{t'}) = \overline{R^{\geq}_B}(X^t) \cup \{x_j\}$
  end if
end for
for $j = n+1$ to $n+n^+$ **do**
  for $j = m+1$ to $m+m^+$ **do**
    compute $[x_i]^{\geq t''}_{\{a_j\}}$;
    $[x_i]^{\geq t+1}_{C^{t+1}} \leftarrow [x_i]^{\geq t}_{C^t} \cap [x_i]^{\geq t''}_{\{a_j\}}$
    if $f(x_i,d) = k$, $x_i \in BNR^{\geq}(X^t)$, $[x_i]^{\geq t+1}_{C^{t+1}} \subseteq X^t$ **then**
      $\underline{R^{\geq}_B}(X^{t+1}) = \underline{R^{\geq}_B}(X^t) \cup \{x_i\}$
    end if
  end for
end for
return $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1})$

$U^t = \{x_i | 1 \le i \le 9\}$, $A^t = \{a_i | 1 \le i \le 3\}$, $U^t/d^t = \{\{x_1,x_2,x_5,x_7,x_9\}, \{x_3,x_4,x_6,x_8\}\}$, $X^t = \{x_1,x_2,x_5,x_7,x_9\}$. $[x_1^{S^t}]_{C^t}^{\succeq} = \{x_1,x_3,x_6\}$, $[x_2^{S^t}]_{C^t}^{\succeq} = \{x_2,x_9\}$, $[x_3^{S^t}]_{C^t}^{\succeq} = \{x_3,x_2,x_5\}$, $[x_4^{S^t}]_{C^t}^{\succeq} = \{x_4,x_6\}$, $[x_5^{S^t}]_{C^t}^{\succeq} = \{x_5,x_1,x_7\}$, $[x_6^{S^t}]_{C^t}^{\succeq} = \{x_3,x_6\}$,

$[x_7^{S^t}]_{C^t}^{\succeq} = \{x_7,x_4,x_2\}$, $[x_8^{S^t}]_{C^t}^{\succeq} = \{x_4,x_6,x_8\}$, $[x_9^{S^t}]_{C^t}^{\succeq} = \{x_6,x_9\}$. $\underline{R_{C^t,S^t}^{\succeq}}(X^t) = \{x_2^{S^t}, x_5^{S^t}\}$, $\overline{R_{C^t,S^t}^{\succeq}}(X^t) = \{x_2^{S^t}, x_5^{S^t}, x_1^{S^t}, x_7^{S^t}, x_9^{S^t}, x_3^{S^t}, x_8^{S^t}\}$, $BNR_{C^t,S^t}^{\succeq}(X^t) = \{x_1^{S^t}, x_7^{S^t}, x_9^{S^t}, x_3^{S^t}, x_8^{S^t}\}$.

Table 7.1. A set-valued information system at time t

U^t	a_1	a_2	a_3	d
x_1	{1}	{2}	{1,2}	0
x_2	{2,3}	{1,2}	{1,3}	0
x_3	{1,2}	{1,2}	{1,2}	1
x_4	{3}	{2,3}	{2,3}	1
x_5	{1}	{2}	{2}	0
x_6	{1,3}	{2,3}	{1,2,3}	1
x_7	{1,2}	{2}	{2}	0
x_8	{3}	{2}	{2,3}	1
x_9	{2,3}	{1,2}	{1,2,3}	0

Now, we consider the case when objects and attributes are added to the set-valued information system simultaneously. The set-valued information system shown in Table 7.1 change to the set-valued information system shown in Table 7.2, where $C^+ = \{a_4, a_5\}$ and $U^+ = \{x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}$. Now the set-valued information system S^{t+1} is decomposed into two set-valued information systems S^{A+}(shown in Table 7.3) and S^{U^+} (shown in Table 7.5). The set-valued information system S^{A^+} is further decomposed into two set-valued information systems S^t (shown in Table 7.1) and $S^{\Delta A}$ (shown in Table 7.4).

Table 7.2. The set-valued information system at time $t+1$

U^{t+1}	a_1	a_2	a_3	a_4	a_5	d
x_1	{1}	{2}	{1,2}	{2}	{1,3}	0
x_2	{2,3}	{1,2}	{1,3}	{3}	{3}	0
x_3	{1,2}	{1,2}	{1,2}	{1,2}	{1,3}	1
x_4	{3}	{2,3}	{2,3}	{3}	{2,3}	1
x_5	{1}	{2}	{2}	{2}	{2,3}	0
x_6	{1,3}	{2,3}	{1,2,3}	{1,2,3}	{1,3}	1
x_7	{1,2}	{2}	{2}	{1,2}	{1,2,3}	0
x_8	{3}	{2}	{2,3}	{3}	{1,2,3}	1
x_9	{2,3}	{1,2}	{1,2,3}	{1,3}	{1,2,3}	0
x_{10}	{1,2,3}	{1,2,3}	{1,2,3}	{1,2,3}	{2,3}	1
x_{11}	{1,2,3}	1,2	1,2,3	2,3	{2,3}	0
x_{12}	{2}	{3}	{1,2,3}	{1,3}	{2}	0
x_{13}	{1,2}	{2}	{2,3}	{2}	{3}	1
x_{14}	{3}	{1,3}	{1,2}	{2,3}	{2}	0

Then, we calculate the approximation incrementally as follows:

1. We compute the dominating classes in the set-valued information system $S^{\Delta A}$. For the set-valued information system $S^{\Delta A}$, $[x_1^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_1, x_3, x_6, x_7\}$, $[x_2^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_2, x_4, x_6, x_8, x_9\}$, $[x_3^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_3, x_6, x_7\}$, $[x_4^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_4, x_8, x_9\}$, $[x_5^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_5, x_7\}$, $[x_6^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_6\}$, $[x_7^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_7\}$, $[x_8^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_8, x_9\}$, $[x_9^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_9\}$.

Table 7.3. The set-valued information system S^{A^+}

U^t	a_1	a_2	a_3	a_4	a_5	d
x_1	{1}	{2}	{1,2}	{2}	{1,3}	0
x_2	{2,3}	{1,2}	{1,3}	{3}	{3}	0
x_3	{1,2}	{1,2}	{1,2}	{1,2}	{1,3}	1
x_4	{3}	{2,3}	{2,3}	{3}	{2,3}	1
x_5	{1}	{2}	{2}	{2}	{2,3}	0
x_6	{1,3}	{2,3}	{1,2,3}	{1,2,3}	{1,3}	1
x_7	{1,2}	{2}	{2}	{1,2}	{1,2,3}	0
x_8	{3}	{2}	{2,3}	{3}	{1,2,3}	1
x_9	{2,3}	{1,2}	{1,2,3}	{1,3}	{1,2,3}	0

2. We compute the dominating classes in the set-valued information system S^{A^+} incrementally. For the set-valued information system S^{A^+}, $[x_1^{S^{A+}}]^{\geq}_{C^{t+1}} = [x_1^{S^t}]^{\geq}_{C^t} \cap [x_1^{S^{\Delta A}}]^{\geq}_{C^+} = \{x_1, x_3, x_6\} \cap \{x_1, x_3, x_6, x_7\} = \{x_1, x_3, x_6\} = [x_1^{S^t}]^{\geq}_{C^t}$, $[x_2^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_2, x_9\} = [x_2^{S^t}]^{\geq}_{C^t}$, $[x_3^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_3\}$, $[x_4^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_4\}$, $[x_5^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_5, x_7\}$, $[x_6^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_6\}$, $[x_7^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_7\}$, $[x_8^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_8\}$, $[x_9^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_9\}$.

Table 7.4. The set-valued information system $S^{\Delta A}$

$U^{\Delta A}$	a_4	a_5	d
x_1	{2}	{1,3}	0
x_2	{3}	{3}	0
x_3	{1,2}	{1,3}	1
x_4	{3}	{2,3}	1
x_5	{2}	{2,3}	0
x_6	{1,2,3}	{1,3}	1
x_7	{1,2}	{1,2,3}	0
x_8	{3}	{1,2,3}	1
x_9	{1,3}	{1,2,3}	0

3. We compute the dominating classes and approximations of a concept in the set-valued information system S^{U^+}. For the set-valued information system S^{U^+}, $[x_{10}^{S^{U+}}]^{\geq}_{C^{t+1}} = \{x_{10}\}$, $[x_{11}^{S^{U+}}]^{\geq}_{C^{t+1}} = \{x_{11}, x_{10}\}$, $[x_{12}^{S^{U+}}]^{\geq}_{C^{t+1}} = \{x_{12}\}$, $[x_{13}^{S^{U+}}]^{\geq}_{C^{t+1}} = \{x_{13}, x_{10}\}$, $[x_{14}^{S^{U+}}]^{\geq}_{C^{t+1}} = \{x_{14}, x_{10}\}$. $S^{U^+} \Big/ d^{U+} = \{\{x_{10}, x_{13}\}, \{x_{11}, x_{12}, x_{14}\}\}$,

Table 7.5. The set-valued information system S^{U^+}

U^+	a_1	a_2	a_3	a_4	a_5	d
x_{10}	{1,2,3}	{1,2,3}	{1,2,3}	{1,2,3}	{2,3}	1
x_{11}	{1,2,3}	1,2	1,2,3	2,3	{2,3}	0
x_{12}	{2}	{3}	{1,2,3}	{1,3}	{2}	0
x_{13}	{1,2}	{2}	{2,3}	{2}	{3}	1
x_{14}	{3}	{1,3}	{1,2}	{2,3}	{2}	0

$X^{U^+} = \{x_{11}, x_{12}, x_{14}\}$. $R^{\geq}_{C^{t+1},S^{U^+}}(X^{U^+}) = \{x_{12}\}$, $\overline{R^{\geq}_{C^{t+1},S^{U^+}}}(X^{U^+}) = \{x_{11}, x_{14}, x_{12}\}$, $BNR^{\geq}_{C^{t+1},S^{U^+}}(X^{U^+}) = \{x_{11}, x_{14}\}$.

4. We compute the dominating classes in the set-valued information system S^{t+1} incrementally. $[x_2^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_2^{S^t}]^{\geq}_{C^{t+1}} \cup [x_{11}^{S^{U^+}}]^{\geq}_{C^{t+1}} = \{x_2, x_9, x_{11}, x_{10}\}$, $[x_{10}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{10}^{S^{U^+}}]^{\geq}_{C^{t+1}}$, $[x_{12}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{12}^{S^{U^+}}]^{\geq}_{C^{t+1}}$, $[x_{13}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{13}^{S^{U^+}}]^{\geq}_{C^{t+1}}$, $[x_{14}^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_{14}^{S^{U^+}}]^{\geq}_{C^{t+1}}$.
5. Finally, we update the approximations of the set-valued information system S^{t+1} incrementally. $x_3^{S^t} \in BNR^{\geq}_{C^t,S^t}(X^t)$, $[x_3^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_3^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_3\} \cap X^t = \emptyset$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) - \{x_3\} = \{x_2, x_5, x_1, x_7, x_9\}$. $x_7^{S^t} \in BNR^{\geq}_{C^t,S^t}(X^t)$, $[x_7^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_7^{S^{A+}}]^{\geq}_{C^{t+1}} = \{x_7\} \subseteq X^t$, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_7\} = \{x_2, x_5, x_7\}$. $[x_2^{S^{t+1}}]^{\geq}_{C^{t+1}} = [x_2^{S^{A+}}]^{\geq}_{C^{t+1}} \cup [x_{11}^{S^{U^+}}]^{\geq}_{C^{t+1}} = \{x_2, x_9, x_{11}, x_{10}\}$, $x_2^{S^t} \in \underline{R^{\geq}_{C^t,S^t}}(X^t)$, $x_{11}^{S^{U^+}} \in BNR^{\geq}_{C^{t+1},S^{U^+}}(X^{U^+})$, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) - \{x_2\} = \{x_5, x_3, x_7\}$. $x_{12}^{S^{U^+}} \in \underline{R^{\geq}_{C^{t+1},S^{U^+}}}(X^{U^+})$, $\underline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \underline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{12}\} = \{x_5, x_3, x_7, x_{12}\}$. $x_{14}^{S^{U^+}} \in BNR^{\geq}_{C^{t+1},S^{U^+}}(X^{U^+})$, $\overline{R^{\geq}_{C^{t+1},S^{t+1}}}(X^{t+1}) = \overline{R^{\geq}_{C^t,S^t}}(X^t) \cup \{x_{14}\} = \{x_2, x_5, x_1, x_7, x_9, x_{14}\}$.

7.6 Performance Analysis

Experiments have been carried out to verify the effectiveness of the algorithms. Data sets used to test the algorithms are listed in Table 7.6. Data sets 1 to 4 are incomplete information system downloaded from UC Irvine Machine Learning Database Repository (www.ics.uci.edu/~mlearn/MLRepository.html). Dataset 5 is a set-valued information system generated by randomly. Data set 6 is from [21], which is a practical conjunctive set-valued decision table from the test for foreign language ability in Shanxi University, China. We perform the experiments on a computer with Intel Core 2 Duo CPU T6500 2.10GHz CPU, 4.0 GB of memory, running Microsoft Windows Vista Home Basic. Algorithms are developed in C#.

Table 7.6. Data description

Data sets		Abbreviation	Rows	Attributes	Classes	Loss rate of attribute values
1	Lung cancer	Lun	32	57	3	0.27%
2	Horse	Horse	300	29	2	18.43678%
3	Shuttle Landing Control	Shuttle	15	8	2	21.66667%
4	housevotes84	HouseV	435	18	2	3.678161%
5	Samples	Samp	400	40	5	no
6	Language test	Lang	50	6	2	no

We compare the efficiency of the incremental algorithm and the non-incremental learning algorithm. Experimental results are listed in Tables 7.7 and 7.8.

Table 7.7. A comparison of efficiency based on the incremental learning algorithm and the non-incremental learning algorithm

Data sets	The number of objects added	The number of attributes added	AUAOAAD(s)	AAUAOAA(s)	Non-incremental updating(s)
Lun	10	24	0.0154933	0.0193596	0.0297983
HorseV	150	13	1.2718853	1.2718853	1.581161
Shuttle	8	2	0.0032563	0.0035347	0.008766
House	100	6	1.5932707	1.3088443	1.7860024
Samp	400	20	7.8289669	8.4810256	8.0401398
Lang	2	30	0.0377443	0.0527268	0.0532438

Table 7.8. A comparison of efficiency between Algorithm AUAOAAD and Algorithm AAUAOAA on "Language test"

The number of objects added	The number of attributes added	AUAOAAD(s)	AAUAOAA(s)	Non-incremental updating(s)
1	1	0.0057276	0.0044171	0.0137274
2	1	0.0106077	0.0052004	0.0062275
3	1	0.0064592	0.006034	0.0067691
4	1	0.0069583	0.0069021	0.007351
10	2	0.012332	0.017197	0.0125285
20	2	0.0296651	0.0308497	0.0332367
30	2	0.0377443	0.0527268	0.0532438

From Table 7.7, we can see the both Algorithms AUAOAAD and AAUAOAA outperform non-incremental updating methods. Furthermore, we carry out experiments on data set "Language test" to verify the efficiency between Algorithm AUAOAAD and Algorithm AAUAOAA. From Table 7.8, we can see when adding objects 1 to 3 and adding 1 attribute, Algorithm AAUAOAA outperforms Algorithm AUAOAAD. In the case of the addition of object 4 and 1 attribute, there is no much

difference of the two algorithms. When adding 10, 20 and 30 objects and adding 2 attributes, Algorithm AUAOAAD outperforms Algorithm AAUAOAA. This is because the cost of Algorithm AAUAOAA to maintain approximations and dominating classes recursively may exceed the cost of Algorithm AUAOAAD. In the incremental updating methods, both Algorithms AUAOAAD and AAUAOAA use original information and data structure, so the efficiency of knowledge discovery improves.

Let $S^t = (U^t, A^t, V^t, f^t)$ denote the set-valued information system at time t. At time $t+1$, objects U^+ and attributes A^+ are added to the set-valued information system simultaneously, *that is*, $S^{t+1} = (U^{t+1}, A^{t+1}, V^{t+1}, f^{t+1})$, where $U^{t+1} = U^t \cup U^+$, $A^{t+1} = C^{t+1} \cup d^t$, $C^{t+1} = C^t \cup C^+$. In Algorithm AUAOA-AD, we decompose the set-valued information system S^{t+1} into two set-valued information systems, *that is*, $S^{U+} = (U^+, A^{t+1}, V^{U+}, f^{U+})$ and $S^{A+} = (U^t, A^{t+1}, V^{A+}, f^{A+})$ and the set-valued information system S^{A+} is further decomposed into two sub set-valued information systems $S^t = (U^t, A^t, V^t, f^t)$ and $S^{\Delta A} = (U^t, A^{\Delta A}, V^{\Delta}, f^{\Delta})$, where $A^{\Delta A} = C^+ \cup d^t$. Computational complexity of Algorithm AUAOAAD is $O((|A^+ + 1)||U^t| + \left|U^{\Delta}\right|^2 \left|A^{t+1}\right| + \left|U^{t+1}\right|)$. The computational complexity of Algorithm AAUAOAA is $O(|U^t|^2 (|A^+| + |U^+|))$. The computational complexity of non-incrementally updating is $O|A^t + 1|\left|U^{t+1}\right|^2$. The complexity of Algorithms AUAOAAD and AAUAOAA are lower than that of non-incremental updating methods.

7.7 Conclusions

The set-valued information systems evolve with time. Objects, attributes and attributes' values may alter. The concept induced by the subset of the universe may vary too. Therefore, the approximations of a concept as well as certain rules and uncertain rules may vary with time. In this chapter, two algorithms of incremental updating method were proposed to maintain approximations when objects and attributes are added simultaneously. In Algorithm AUAOAAD, we decomposed the set-valued information system into three sub set-valued information systems considering attributes and objects change simultaneously. In Algorithm AAUAOAA, a accumulation policy is taken into consideration. We investigated the variation of approximations related to the variation of granules in the different regions. Then, we presented the methods to update approximations of a concept. An example was employed to explain the AUAOAAD method. Experimental results showed the both methods were effective to maintain knowledge. In the future, we will study optimization techniques of the proposed methods and extend these methods to fuzzy rough set model.

Acknowledgements. This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117), Youth Social Science Foundation of the Chinese Education Commission (10YJCZH117, 11YJC630127) and the Fundamental Research Funds for the Central Universities (SWJTU11ZT08).

References

1. Bilski, P., Wojciechowski, J.M.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99(1), 48–57 (1997)
2. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic variable consistency rough set approaches. International Journal of Approximate Reasoning 50(7), 979–999 (2009)
3. Chen, H.M., Li, T.R., Qiao, S.J., Ruan, D.: A rough sets based dynamic maintenance approach for approximations in coarsening and refining attribute values. International Journal of Intelligent Systems 25, 1005–1026 (2010)
4. Chen, H.M., Li, T.R., Zhang, J.B.: A Method for incremental updating approximations when objects and attributes vary with time. In: Proceedings of 2010 IEEE International Conference on Granular Computing, Silicon Valley, pp. 90–95. IEEE Computer Society Press, Washington, DC (2010)
5. Chen, H.M., Li, T.R., Zhang, J.B.: A Method for Incremental updating approximations based on variable precision set-valued ordered information systems. In: Proceedings of the 2010 IEEE International Conference on Granular Computing, Silicon Valley, pp. 96–101. IEEE Computer Society Press, Washington, DC (2010)
6. Cheng, Y.: The incremental method for fast computing the rough fuzzy approximations. Data & Knowledge Engineering 70(1), 84–100 (2011)
7. Ciucci, D.: Classification of Dynamics in Rough Sets. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 257–266. Springer, Heidelberg (2010)
8. Ciucci, D.: Temporal Dynamics in Rough Sets Based on Coverings. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 126–133. Springer, Heidelberg (2010)
9. Ciucci, D.: Attribute Dynamics in Rough Sets. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 43–51. Springer, Heidelberg (2011)
10. Fan, Y.N., Tseng, T.L., Chern, C.C., Huang, C.C.: Rule induction based on an incremental rough set. Expert Systems with Applications 36(9), 11439–11450 (2009)
11. Guan, Y.Y., Wang, H.K.: Set-valued information systems. Information Sciences 176(17), 2507–2525 (2006)
12. Deng, D., Huang, H.-K.: Dynamic Reduction Based on Rough Sets in Incomplete Decision Systems. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 76–83. Springer, Heidelberg (2007)
13. Hassanien, A.E., Abraham, A., Peters, J.F., Schaefer, G., Henry, C.: Rough sets and near sets in medical imaging: A review. IEEE Transactions on Information Technology in Biomedicine 13(6), 955–968 (2009)
14. Kryszkiewicz, M.: Rough set approach to incomplete. Information Sciences 112, 39–49 (1998)
15. Liu, D., Li, T.R., Ruan, D., Zou, W.L.: An incremental approach for inducing knowledge from dynamic information systems. Fundamenta Informaticae 94, 245–260 (2009)
16. Li, T.R., Ruan, D., Wets, G., Song, J., Xu, Y.: A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. KnowledgeBased Systems 20(5), 485–494 (2007)
17. Pattaraintakorn, P., Cercone, N.: Integrating rough set theory and medical applications. Applied Mathematics Letters 21(4), 400–403 (2008)
18. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)

19. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99(1), 48–57 (1997)
20. Qian, Y.H., Liang, J.Y., Pedrycz, W., Dang, C.Y.: Positive approximation: An accelerator for attribute reduction in rough set theory. Artificial Intelligence 174(9-10), 597–618 (2010)
21. Qian, Y.H., Dang, C.Y., Liang, J.Y., Tang, D.W.: Set-valued ordered information systems. Information Sciences 179(16), 2809–2832 (2009)
22. Skowron, A., Stepaniuk, J., Swiniarski, R.: Approximation spaces in rough-granular computing. Fundamenta Informaticae 100, 141–157 (2010)
23. Song, X.X., Zhang, W.X.: Knowledge reduction in inconsistent set-valued decision information system. Computer Engineering and Applications 45(1), 33–35 (2009)
24. Stefanowski, J., Tsoukiàs, A.: On the Extension of Rough Sets under Incomplete Information. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 73–82. Springer, Heidelberg (1999)
25. Swiniarski, R.W., Pancerz, K., Suraj, Z.: Prediction of model changes of concurrent systems described by temporal information systems. In: Hamid, R. (ed.) Proceedings of the 2005 International Conference on Data Mining, Las Vegas, Nevada, USA, pp. 51–57. CSREA Press (2005)
26. Wang, G.Y.: Extension of rough set under incomplete information systems. Journal of Computer Research and Development 39(10), 1238–1243 (2002)
27. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
28. Zheng, Z., Wang, G.Y.: RRIA: A rough set and rule tree based incremental knowledge acquisition algorithm. Fundamenta Informaticae 59, 299–313 (2004)
29. Zou, W.L., Li, T.R., Chen, H.M., Ji, X.L.: Approaches for incrementally updating approximations based on set-valued information systems while attribute values' coarsening and refining. In: Proceedings of the 2009 IEEE International Conference on Granular Computing, pp. 824–829. IEEE Computer Society Press, Washington, DC (2009)

Chapter 8
On the Gradual Evolvement of Things

Ivo Düntsch and Günther Gediga

Abstract. The aim of this work is to describe the generic properties of a visual system without hard coding the environment.

Keywords: hard coding the environment, indicator function, invariants, perceptional vector analysis, quadtrees, rough approximation quality, translations, visual perception.

8.1 Introduction

Over a period of fifty years, the psychologist J. J. Gibson developed a theory of visual perception that was radically different from the prevailing views of the time - and, to some extent, from those of today. According to Gibson [5], the main task for perception is to recognize the invariant parts within a variant world:

> "We perceive that the environment changes in some respects and persists in others. We see that it is different from time to time, even moment to moment, and yet that it is the same environment over time. We perceive *both* the change and the underlying non–change. My explanation is that the perceptual systems work by detecting invariants in the flux of stimulation but are also sensitive to the flux itself" [4].

These invariants can be described by a mathematical model, which can be used more than once for the description of the world. Gibson showed that simple models

Ivo Düntsch
Brock University, St. Catharines, Ontario, Canada, L2S 3A1
e-mail: duentsch@brocku.ca

Günther Gediga
Department of Psychology, Institut IV, Universität Münster, Fliednerstr. 21,
Münster, Germany
e-mail: gediga@uni-muenster.de

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 249–259.
springerlink.com

define properties (*e.g.,* "horizon"), but these models are rather ad hoc, and a general framework is still lacking.

A principle of visual perception based on an ecological approach was outlined by Johansson, which he called *perceptional vector analysis* [6]. Johansson showed by experiments that visual perception seems to be organized by something close to factoring independent vectors of movement patterns. A simple experiment was the presentation of three light points in a dark environment. Two points are moving up and down, the third (inner) point is moving up and down as well, but moving at the same time gradually from one outer point to the other. The perceptional result is as follows: There is a "stick" moving up and down (the two outer points) and a point moving up and down on the stick.

The experimental results show that the perceptual content is independent of an observer, although the layout consists of three moving points only. Johannson concluded that the visual system is organized in such a way that this certain layout has to be perceived as a hierarchical system of – more or less independent – movement patterns. In this chapter, we aim at a description of a perceiving system from an algorithmic/logical point of view, which is more general than the assumption of a system that performs vector analysis. Furthermore, we like to deal with an imprecise information basis, which we describe by rough sets.

Much effort has been put into the mathematical description of spatial or spatial-temporal relationships, which use logical/algebraical models such as the *region connection calculus* [10], the *polygonal algebras* [9], and *contact* or *proximity algebras* [2]. These models use "regions" as primitives. According to Gibson, objects should be defined by invariants; however, in the models mentioned above, the nature of these invariants is not addressed, since regions are treated as existing a priori.

In a predefined world, shapes of objects such as blocks, cones, disks are hard coded into the system. Through detectors, the agent discovers objects in its environment according to its instructions. In this chapter, we aim to describe the generic properties of a visual system without hard coding the environment. As a measure of approximation, we will use Pawlak's approximation quality [8].

8.2 The Setup

In the first approach, we use instances $t_1, t_2, \ldots$ of an $n \times n$ grid $C = \{c_{i,j} : i, j < n\}$ of cells, called a *container*, which carries the information coming from the outside world to an organism or a machine. We may think of the cells as atoms of a two dimensional visual field spanned by C, and each cell contains information delivered by a sensor. The state of the cell $c_{i,j}$ at time t_k is denoted by $[c_{i,j}]_k$ and is evaluated according to attributes such as color, hue, intensity, or just "on" or "off"; indeed, each cell may contain a whole information system in the sense of [7]. The global state of C at time t_k is denoted by C_k.

The *indicator function* measures the amount of agreement between two global states at times t_r and t_s:

$$\mathrm{ind}_{r,s}(c_{i,j}) = \begin{cases} 1, & \text{if } [c_{i,j}]_r = [c_{i,j}]_s \\ 0, & \text{otherwise,} \end{cases} \tag{8.1}$$

Its complementary xor function is defined by

$$\mathrm{xor}_{r,s}(c_{i,j}) = \begin{cases} 1, & \text{if } [c_{i,j}]_r \neq [c_{i,j}]_s, \\ 0, & \text{otherwise.} \end{cases}$$

The (global) amount of agreement in states t_r and t_s is measured by $\sum\{\mathrm{ind}_{r,s}(c_{i,j}) : i,j < n\}$, denoted by $I_{r,s}$, and the (global) amount of change from t_r to t_s is measured by $\sum\{\mathrm{xor}_{r,s}(c_{i,j}) : i,j < n\}$, denoted by $D_{r,s}$.

A *movement* is a mapping $m : C \to C \cup \{\infty\}$ which is (partly) injective in the following sense:

$$\text{If } \langle i,j\rangle \neq \langle i',j'\rangle, \text{ and } \{m(c_{i,j}), m(c_{i',j'})\} \neq \{\infty\}, \text{ then } m(c_{i,j}) \neq m(c_{i',j'}). \tag{8.2}$$

We interpret $m(c_{i,j}) = \infty$ as "cell $c_{i,j}$ moves out of sight".

With each movement m and two states t_r and t_s we can compare $[c_{i,j}]_r$ and $[m(c_{i,j})]_s$, and set

$$\mathrm{ind}^m_{r,s}(c_{i,j}) = \begin{cases} 1, & \text{if } [c_{i,j}]_r = [m(c_{i,j})]_s, \\ 0, & \text{otherwise.} \end{cases} \tag{8.3}$$

The agreement with respect to m is defined by

$$I^m_{r,s} := \sum\{\mathrm{ind}^m_{r,s}(c_{i,j}) : i,j < n\}. \tag{8.4}$$

For the corresponding difference function, we need to take into account that cells may move outside the visual field: If $m(c_{i,j}) \neq \infty$, set

$$\mathrm{xor}^m_{r,s}(c_{i,j}) = \begin{cases} 1, & \text{if } [c_{i,j}]_r \neq [m(c_{i,j})]_s, \\ 0, & \text{otherwise.} \end{cases} \tag{8.5}$$

The difference index $D^m_{r,s}$ is given by

$$D^m_{r,s} := \sum\{\mathrm{xor}^m_{r,s}(c_{i,j}) : m(c_{i,j}) \neq \infty\}. \tag{8.6}$$

Since C is finite, so is the set $\mathcal{M}$ of movements, and therefore, $\max\{I^m_{r,s} : m \in \mathcal{M}\}$ and $\min\{I^m_{r,s} : m \in \mathcal{M}\}$ are well defined.

8.3 Translations

Simple examples of movements are translations $t(x,y)$, $x,y \in Z$ with

$$t(x,y)(c_{i,j}) = \begin{cases} c_{i+x,j+y}, & \text{if } 0 \leq i+x, j+y < n, \\ \infty, & \text{otherwise.} \end{cases}$$

For any $x,y < n$, the value $I^{x,y}_{r,s}$ offers us an evaluation of a translation in coordinates given by C. The parameters x,y of a maximal value of $I^{x,y}_{r,s}$ is a candidate for the representation of an outer world translation by internal coordinates assuming a "stable world" assumption.

Example 8.1. Assume the following two representations at time $r = 1$ and $s = 2$ using a grid with $n = 5$ with two "outer points" and one "inner point" just like the layout in Johannson's experiment:

```
r = 1          s = 2
XX00X          00000
XX00X          00000
00000          00000
00000          X0X0X
00000          X0X0X
```

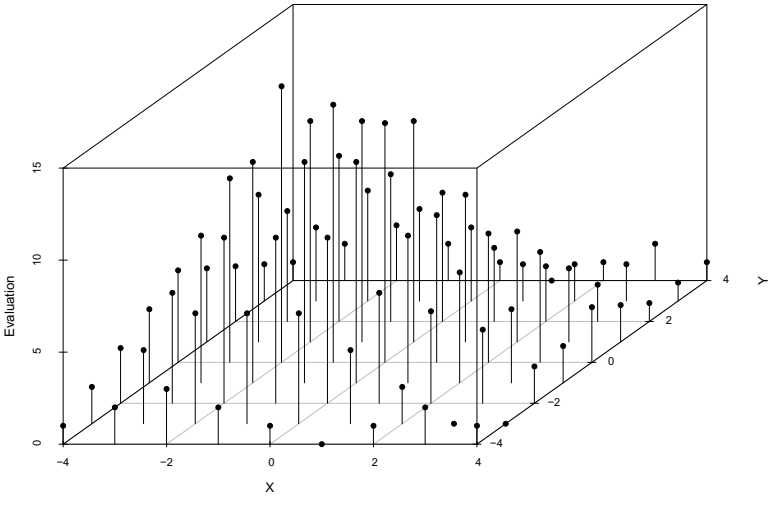


Fig. 8.1. $I^{x,y}_{1,2}$, $x,y \in \{-4,...,+4\}$

Evaluating all values of $I_{1,2}^{x,y}$ using $x,y \in \{-4,...,+4\}$ results in the functions shown in Figure 8.1. In this situation, there is a maximal value of 15 which is unique, since 15 elements can be recaptured if and only if $x = -2$ and $y = 0$. In other words, the translation $t(-2,0)$ results in the greatest agreement.

Note that, the number of elements can be rescaled in terms of rough approximation quality

$$\lambda_0 = \frac{\text{recaptured elements by translation}}{\text{elements of the grid}}$$

The index 0 is used, because λ_0 tells us something about the overall approximation quality of background and translation stable things. In our example, we observe $\lambda_0 = 0.6$.

In order to extract more translations from the grids, we mark with asterisks those cells $c_{i,j}$ in C_1 and $t(-2,0)(c_{i,j})$ in C_2 for which $[c_{i,j}]_1 = [t(-2,0)(c_{i,j})]_2$.

```
r = 1          s = 2
XX00X          *****
XX00X          *****
*****          *****
*****          X0X0X
*****          X0X0X
```

We call these the *residual grids* $C_1(1)$ *and* $C_2(1)$. Now we apply the maximization algorithm once again to the non–asterisk cells, resulting in the optimal value of 6 at $(x = 3, y = 0)$.

Once again, the approximation quality may be defined by

$$\lambda_1 = \frac{\text{recaptured remaining elements by translation}}{\text{remaining elements of the grid}},$$

resulting in $\lambda_1 = 0.6$, which indicates that 60% of the remaining area can be recaptured by a translation, which is different from the background movement.

Removing those elements that were recaptured by this step results in

```
r = 1          s = 2
*X0**          *****
*X0**          *****
*****          *****
*****          *0X**
*****          *0X**
```

Now – at step 2 – the situation is not unique(Figure 8.3), because two translations of $C_1(2)$ to $C_2(2)$ offers identical local optimum values, which is quite probable for low frequencies. Whatever translation will be used, we result in $\lambda_2 = 0.5$.

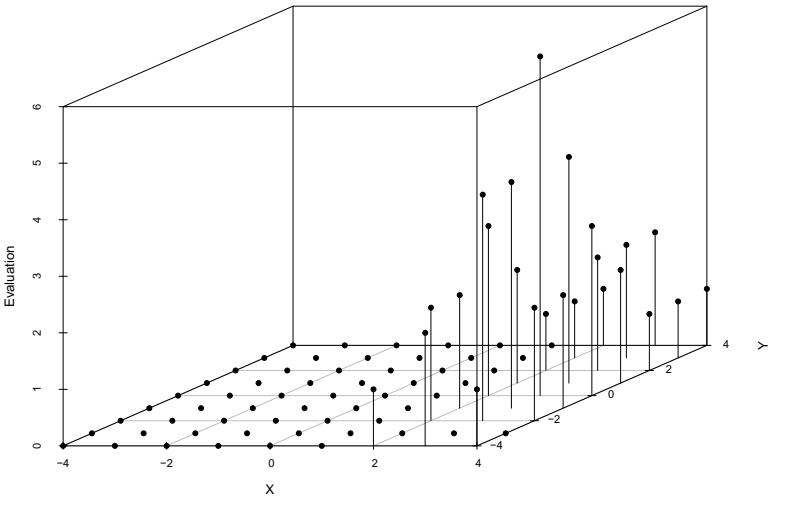


Fig. 8.2. $I_{1,2}^{x,y},\ x,y \in \{-4,...,+4\}$ for residual grids I

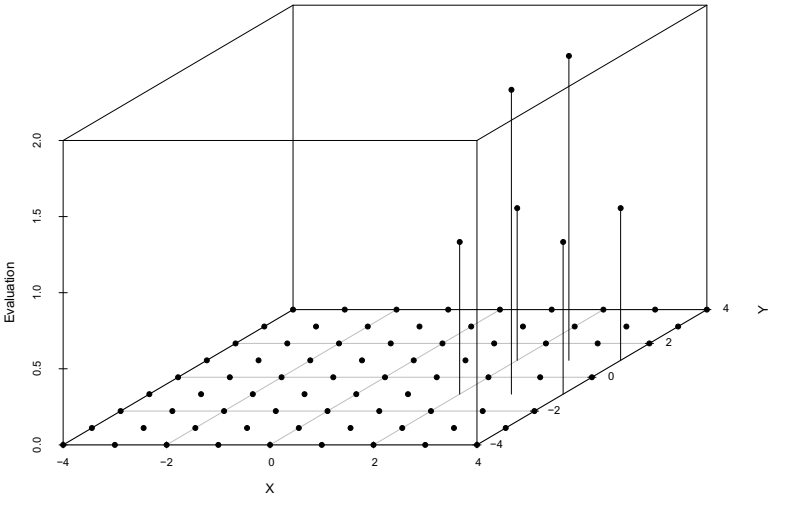


Fig. 8.3. $I_{1,2}^{x,y},\ x,y \in \{-4,...,+4\}$ for residual grids II

Nevertheless, if we choose one of the arguments of two optimal values, *for example*, $x = 3, y = 1$, we end up in a movement of two elements within those 6 elements of step 1.

□

Obviously, the algorithm can be applied recursively until either grid $C_1(k)$ or $C_2(k)$ is empty.

The rough set analysis of translations offers the same results as Johansson's perceptual vector theory: The elements show a moving stick (recaptured in step 1) and an element moving within the stick (recaptured in step 2).

However, up to now, we do not have a description of a "thing". We will explore in the next section how to define things in terms of a stable residual.

8.3.1 Evolving Things and Properties

The situation with two successive frames offers a description of "T-things" by collecting at each step those cells whose content agrees with the content of its image. We result in 4 different T-things that consists of an *s*-position, a set of 2-dimensional values and a direction due to the translation.

Our grid at $s = 2$ can be translated into regions defined by T-things:

```
s = 2
11111
11111
11111
24322
24322
```

The values of these 4 T-things are

T-thing 1 ($Tt = 1$)

```
00000
00000
00000
```

T-thing 1 is a block consisting of "0" elements. Obviously, we result in something like a block property, which a system can learn and add this to the possible attributes of a (more complex than a T-) thing.

T-thing 2 ($Tt = 2$)

```
X??0X
X??0X
```

T-thing 2 consists of "X" elements at the end of the block and "0" elements at y position within the block. The question marks show cells, which are not part of T–thing 2, but T–things further down in the hierarchy. So these positions may be filled

in by other T-things that may define a more complex "composite T–Thing", because the positions of the subsequent T–things are "within" the region of T–thing 2.

T-thing 3 ($Tt = 3$)

```
X
X
```

T-thing 4 ($Tt = 4$)

```
0
0
```

Finally, every T-thing has its own direction of movement, which may be used to predict the results of the following frame 3.

The expectation of the visual situation of frame 3 shows the following pattern:

```
00000
?????
?????
?????
?????     (end of the visual field)
?????
a??bX
a??bX
```

a = either X (T-thing 2) or 0 (T-thing 4)
? = no clue at all
b = either 0 (T-thing 2) or 0 (T-thing 3)

So the system "now" which will be in sight at frame 3 and it has some further predictions even for results out of sight.

Assume for a moment that T-thing 2 is in sight at frame 3.

- If region "a" consists of X only, the system may conclude that T-thing 2 is in front of T-thing 4.
- If region "a" consists of 0 only, the system may conclude that T-thing 4 is in front of T-thing 2.
- If region "b" consists of X only, the system may conclude that T-thing 3 is in front of T-thing 2.
- If region "b" consists of 0 only, the system may conclude that T-thing 2 is in front of T-thing 3.

We conclude with the observation that when using 3 frames there is a chance of detecting an occlusion relation among T-things.

8.4 Quadtrees

Using a pyramid representation for data is quite old [1]. The hierarchical organization of visual input has two nice properties:

- Computation within higher layer of a hierarchical organized visual input is much faster than dealing with the basic elements.
- Good organization in a higher level helps to organize selective attention [11].

Our idea is the organization of the visual field in quadtrees. These data structures have the nice property that building the structure is based on a rough set operation. Loosely speaking, a quadtree [3] consists of a succession of decompositions of a given rectangle C into quadrants until each node satisfies a given condition. The process can be represented by a tree, each non–leaf node of which has four children, corresponding to the quadrants NW, NE, SW, and SE; in matrix form,

$$\begin{pmatrix} a^k_{00} & a^k_{01} \\ a^k_{10} & a^k_{11} \end{pmatrix}$$

Each node has either no descendants or four descendants. Each descendant of a node g represents a quadrant in the plane whose origin is g and which is bounded by the quadrant boundaries of the previous step; an example taken from [12] is shown in Figure 8.4.

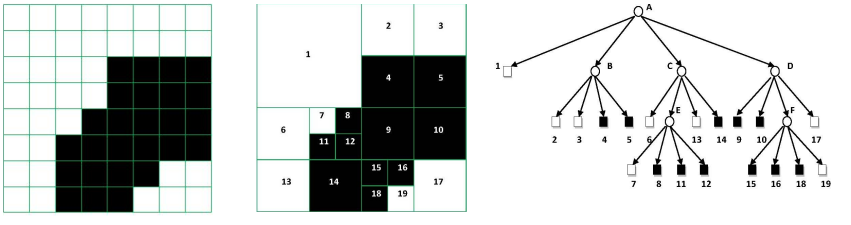


Fig. 8.4. A region I of cells, its decomposition, and its quadtree representation

Each tree level corresponds to an equivalence relation on C. A node becomes a leaf node if its associated quadrant – that is, its corresponding equivalence class – is uni–colored; otherwise, the equivalence relation will be refined to the next level. This is the rough set concept of nested equivalence relations.

Representations consisting of two or more quadtrees can be used to analyze movements as well. The detection of a global translation in the basic layout (bottom of the quadtree) will be very time consuming. A detection of global translation is easier, if we compare the quadtree representation at a higher level. For example, suppose block 1 in Figure 8.4 is black, and all other blocks are white. In this case, we need only 4 bits to represent the full layout. Assume additionally that block 1 moves to one of the other three possible blocks in that quadtree representation. In this case, we need only a few operations to detect the global movement, based on the equivalence classes in the top layer of the quadtree representation.

8.5 Conclusion and Outlook

Although the ideas presented here are far away from a comprehensive theory, we demonstrate that simple movements – translations – can be detected by a recursive rough set based (counting) algorithm. Step 0 will always be the detection of (mutual!) global movement based on the best rough description of the visual grid. Filtering out the results of this step, we end up in a generic hierarchical description of movements, based on rough approximations of residual sets. The results of this procedure lead – at least for our examples – to the same results as the perceptual vector theory of ecological psychologists.

The spirit of the ecological approach – that essential information about "things" is given by their movement behavior – is applied to a very simple layout. No pre–assumption is necessary to detect movement, up to the generic idea that the world is invariant if we take certain transformations (such as translations) into account. We demonstrate that the optimization can be applied to detect "T-things" which will (or will not) be real things, but which can be characterized by their common (residual) movement. We have demonstrated that even concepts such as "occlusion" can be defined on this basis: The information of an "occlusion" is given directly by the changing environment and the invariance assumption(s) of the perceiving system.

We like to point out that "T-things" are not "things" in colloquial parlance, but may be "things" if they are sufficiently interesting for the organism/agent/robot. The definition of what is "interesting" for an organism is based on the concept of affordances by Gibson. Conceptually, we have to define a new representation of "T-things" enhanced by some value for the organism based upon one or more affordances, which we will call "things" from now on. Consequently, whereas a "T-thing" will vanish, if it is out of sight (because there is no additional value for the organism), a "T-thing", enhanced by some affordances, may persist within a different data structure. Using this kind of "thing" evolvement, there will be no "hole" in the perceptional world if a "T-thing" vanishes: either it is of no interest – then other "T-things" step into the perception – or it is of some interest; in this case the organism is aware that the "thing" has moved outside the visual field, with a certain gradient and the organism can capture the "thing" by moving itself to focus the visual field on the thing once again.

The final point of our chapter deals with the fact that detecting change might be quite time consuming, if the visual field is not as tiny as in our examples. In that case, a further rough set based information structure can be used: The quadtree representation of the visual field.

Since this chapter presents only the first ideas, the validity of the approach is still an open question and needs to be tested. One has to define algorithms that can handle huge visual fields, and we have to consider other types of movements like rotation. Nevertheless, a construction of a system which is flexible to define its own "reality" based on perceived "things" is possible and seems to be a challenge for the future.

Acknowledgements. We would like to thank the referee for the helpful and constructive comments.

References

1. Burt, P.: Smart Sensing within a Pyramid Vision Machine. Proceedings of the IEEE 76, 1006–1015 (1988)
2. Dimov, G., Vakarelov, D.: Contact algebras and region–based theory of space: A proximity approach – I,II. Fundamenta Informaticae 74, 209–282 (2006)
3. Finkel, R., Bentley, J.: Quad trees: A data structure for retrieval on composite keys. Acta Informatica 4, 1–9 (1974)
4. Gibson, J.J.: On the new idea of persistence and change and the old idea that it drives out. In: Reed, E., Jones, R. (eds.) Reasons for Realism – Selected Essays of James J. Gibson, pp. 393–396. Lawrence Erlbaum Associates, Hillsdale (1982)
5. Gibson, J.J.: The Ecological Approach to Visual Perception. Lawrence Erlbaum, Hillsdale (1986)
6. Johannson, G.: Visual perception of biological motion and a model for its analysis. Perception & Psychophysics 14, 201–211 (1974)
7. Pawlak, Z.: Information systems, theoretical foundations. Information Systems 6, 205–218 (1981)
8. Pawlak, Z.: Rough sets. Internat. J. Comput. Inform. Sci. 11, 341–356 (1982)
9. Pratt, I., Schoop, D.: A complete axiom system for polygonal mereotopology of the real plane. Journal of Philosophical Logic 27(6), 621–658 (1998)
10. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Nebel, B., Swartout, W., Rich, C. (eds.) Proc. 3rd Int. Conf. Knowledge Representation and Reasoning, pp. 165–176. Morgan Kaufmann (1992)
11. Tsotsos, J., Culhane, S., Cutzu, F.: From theoretical foundations to a hierarchical circuit for selective attention. In: Braun, J., Koch, C., Davis, J. (eds.) Visual Attention and Cortical Circuits, pp. 285–306 (2001)
12. Yin, X., Düntsch, I., Gediga, G.: Quadtree representation and compression of spatial data. Transactions on Rough Sets 13, 207–238 (2010)

Chapter 9
An Empirical Comparison of Rule Sets Induced by LERS and Probabilistic Rough Classification

Jerzy W. Grzymała-Busse, Shantan R. Marepally, and Yiyu Yao

Abstract. We explore an extension of rough set theory based on probability theory. Lower and upper approximations, the basic ideas of rough set theory, are generalized by adding two parameters, denoted by alpha and beta. In our experiments, for different pairs of alpha and beta, we induced three types of rules: positive, boundary, and possible. The quality of these rules was evaluated using ten-fold cross-validation on five data sets. The main results of our experiments are that there is no significant difference in quality between positive and possible rules and that boundary rules are the worst.

Keywords: Rough set theory, Pawlak approximations, probabilistic approximations, parameterized approximations, positive, boundary and possible rules.

9.1 Introduction

Rough set theory, introduced by Z. Pawlak in the early eighties of the last century [7, 8], is widely applied in many research areas. One of the most successful application areas is data mining.

Jerzy W. Grzymała-Busse
Department of Electrical Engineering and Computer Science University of Kansas, Lawrence, KS 66045 USA and Institute of Computer Science, Polish Academy of Science, 01-237 Warsaw Poland
e-mail: jerzy@ku.edu

Shantan R. Marepally
Department of Electrical Engineering and Computer Science University of Kansas, Lawrence, KS 66045 USA
e-mail: shantan@ku.edu

Yiyu Yao
Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2
e-mail: yyao@cs.uregina.ca

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 261–276.
springerlink.com

Many extensions of rough set theory were investigated, one of such extensions is based on probability theory. There exist a few approaches to extending rough set theory using probability theory. One of them, used in this paper, is a decision-theoretic rough set model (DTRSM) in which a pair of threshold parameters for defining probabilistic approximations was determined based on the well established Bayesian decision theory [13, 15, 16]. The probabilistic approximations defined by such parameters would incur minimal risk in deciding the positive, boundary, and negative regions. Based on intuitive arguments, a variable precision rough set model (VPRSM) for probabilistic approximations was suggested in [18]. Once probabilistic approximations are introduced, it is possible to induce rules using standard methodology [6, 10, 12, 5].

For the rest of this chapter, we will assume that the knowledge is represented by rules and that data sets are represented by decision tables. Additionally, this chapter, is a continuation of our previous work [5] in which it was shown how to modify decision tables in order to induce positive, boundary, and possible rules induced from probabilistic approximations. However, this time, we simplify the idea of modification of decision tables from which such rules may be induced. Furthermore, we present results of experiments on typical data sets from the University of California at Irvine Machine Learning Repository. These data sets were additionally processed before our experiments. Namely, during discretization, intervals for numerical attributes were allowed to be too large so that the data sets were inconsistent. This additional modification of data sets was necessary since with an appropriate discretization and consistent data sets lower and upper probabilistic approximations will be identical.

A preliminary version of this chapter was prepared for the 10-th International Conference on Hybrid Intelligent Systems (HIS 2010), Atlanta GA, August 23–25, 2010 [4].

9.2 Rough Sets and Three-Way Rules

In this section, the notion of three-way rules is introduced with respect to Pawlak and probabilistic rough sets.

9.2.1 Indiscernibility Relation

As stated before, we assume that the input data sets are presented in the form of a *decision table*. An example of a decision table is shown in Table 9.1.

Rows of the decision table represent *cases*, while columns are labeled by *variables*. The set of all cases will be denoted by U. In Table 1, $U = \{1, 2,..., 10\}$. Independent variables are called *attributes*, and a dependent variable is called a *decision* and is denoted by d. The set of all attributes will be denoted by A. In Table 1, $A = \{Temperature, Headache, Cough\}$. The value for a case x and an attribute a will be denoted by $a(x)$.

One of the most important ideas of rough set theory is an indiscernibility relation. Let Q be a non-empty subset of A. The indiscernibility relation $IND(Q)$ is a relation on U defined for $x, y \in U$ as follows:

$$(x, y) \in IND(Q) \text{ if and only if } \forall a \in Q(a(x) = a(y)).$$

The indiscernibility relation $IND(Q)$ is an equivalence relation. Equivalence classes of $IND(Q)$ are called *elementary sets* of Q and are denoted by $[x]_Q$. A subset of U is called *Q-definable* if it is a union of elementary sets [7, 8].

Table 9.1. A decision table

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	normal	no	no	no
2	normal	no	no	no
3	normal	yes	no	no
4	normal	yes	no	no
5	normal	yes	no	yes
6	normal	no	yes	no
7	high	yes	yes	no
8	high	yes	yes	yes
9	high	yes	yes	yes
10	high	yes	yes	yes

9.2.2 Pawlak Approximations

The set C of all cases defined by the same value of the decision d is called a *concept*. Usually, a concept C is not Q-definable. The largest definable set contained in C is called the lower approximation of C, and the smallest definable set containing C is called the upper approximation of C, namely,

$$\begin{aligned} \underline{apr}_Q(C) &= \bigcup \{[x]_Q \mid [x]_Q \subseteq C\} \\ &= \bigcup \{[x]_Q \mid Pr(C|[x]_Q) = 1\}; \\ \overline{apr}_Q(C) &= \bigcup \{[x]_Q \mid [x]_Q \cap C \neq \emptyset\} \\ &= \bigcup \{[x]_Q \mid Pr(C|[x]_Q) > 0\}, \end{aligned} \tag{9.1}$$

where $Pr(C|[x]_Q) = |C \cap [x]_Q|/|[x]_Q|$ is the conditional probability, and $|\cdot|$ is the cardinality of a set. The set $\text{POS}_Q(C) = \underline{apr}_Q(C)$ is called the positive region of C,

the set $\mathrm{BND}_Q(C) = \overline{apr}_Q(C) - \underline{apr}_Q(C)$ is called the boundary region of C, and the set $\mathrm{NEG}_Q(C) = U - \overline{apr}_Q(C)$ is called the negative region of C.

For an attribute a and its value v, (a,v) is called an attribute-value pair. The set of objects, $[(a,v)] = \{o \in U \mid a(o) = v\}$, is called a block [2]. For Table 9.1, the concept of Flu is defined by the block $[(Flu,yes)]$, and the concept of $No\text{-}Flu$ is defined by the block $[(Flu,no)]$. For $Q = A$, the lower approximations of both concepts are:

$$\underline{apr}_A([(Flu,no)]) = \{1,2,6\},$$
$$\underline{apr}_A([(Flu,yes)]) = \emptyset,$$

and upper approximations of the concepts are:

$$\overline{apr}_A([(Flu,no)]) = U,$$
$$\overline{apr}_A([(Flu,yes)]) = \{3,4,5,7,8,9,10\}.$$

The associated three regions are:

$$\mathrm{POS}_A([(Flu,no)]) = \{1,2,6\},$$
$$\mathrm{BND}_A([(Flu,no)]) = \{3,4,5,7,8,9,10\},$$
$$\mathrm{NEG}_A([(Flu,no)]) = \emptyset,$$

$$\mathrm{POS}_A([(Flu,yes)]) = \emptyset,$$
$$\mathrm{BND}_A([(Flu,yes)]) = \{3,4,5,7,8,9,10\},$$
$$\mathrm{NEG}_A([(Flu,yes)]) = \{1,2,6\}.$$

9.2.3 Probabilistic Approximations

Probabilistic rough sets have been proposed and studied in [9, 12, 13, 15, 16, 18]. In probabilistic approaches to rough sets, such as decision-theoretic model [13, 15] and variable precision model [18], we define the parameterized approximations of the concept C:

$$\underline{apr}_{Q,(\alpha,\beta)}(C) = \bigcup\{[x]_Q \mid Pr(C|[x]_Q) \geq \alpha\},$$
$$\overline{apr}_{Q,(\alpha,\beta)}(C) = \bigcup\{[x]_Q \mid Pr(C|[x]_Q) > \beta\}, \tag{9.2}$$

with $\alpha > \beta$. They are referred to as the α-level lower approximation and β-level upper approximation, respectively. Similarly, (α,β)-level positive, (α,β)-level boundary and (α,β)-level negative regions may be introduced. Additionally, a union of the (α,β)-level positive and (α,β)-level boundary regions is called a (α,β)-level possible region.

Suppose we set $\alpha = 0.6$ and $\beta = 0.3$, which can be interpreted in terms of cost or risk of classifying an object into the positive region, the boundary region or the negative region [13, 15]. For Table 9.1, with respect to $Q = A$. we have the following probabilistic $(0.6, 0.3)$-approximations:

$$\begin{aligned}
\underline{apr}_{A,(0.6,0.3)}([(Flu,no)]) &= \{1,2,3,4,5,6\},\\
\underline{apr}_{A,(0.6,0.3)}([(Flu,yes)]) &= \{7,8,9,10\},\\
\overline{apr}_{A,(0.6,0.3)}([(Flu,no)]) &= \{1,2,3,4,5,6\},\\
\overline{apr}_{A,(0.6,0.3)}([(Flu,yes)]) &= \{3,4,5,7,8,9,10\}.
\end{aligned}$$

The associated positive, boundary and negative regions are:

$$\begin{aligned}
\text{POS}_{A,(0.6,0.3)}([(Flu,no)]) &= \{1,2,3,4,5,6\},\\
\text{BND}_{A,(0.6,0.3)}([(Flu,no)]) &= \emptyset,\\
\text{NEG}_{A,(0.6,0.3)}([(Flu,no)]) &= \{7,8,9,10\},
\end{aligned}$$

$$\begin{aligned}
\text{POS}_{A,(0.6,0.3)}([(Flu,yes)]) &= \{7,8,9,10\},\\
\text{BND}_{A,(0.6,0.3)}([(Flu,yes)]) &= \{3,4,5\},\\
\text{NEG}_{A,(0.6,0.3)}([(Flu,yes)]) &= \{1,2,6\}.
\end{aligned}$$

9.2.4 Three-Way Rules

The four regions may be interpreted in terms of some decisions [14], with the positive region representing the certain acceptance, boundary region representing deferment (i.e., delayed decision or indecision), negative region representing rejection, and possible region representing possible (or plausible) acceptance. Rules induced from the positive, boundary, and possible regions are, respectively, referred to as *positive*, *boundary*, and *possible* rules.

We can analyze three-way rules and examine their implications by introducing a few quantitative measures. The Pawlak approximations may be interpreted as a special case with $\alpha = 1$ and $\beta = 0$. We therefore only consider the probabilistic case.

Consider first two quantitative measures on characterizing an arbitrary set $G \subseteq U$ with respect to a concept $C \subseteq U$. The precision or accuracy of G is given by:

$$precision(C|G) = \frac{|C \cap G|}{|G|}, \tag{9.3}$$

and the recall provided by G for C is given by:

$$recall(C|G) = \frac{|C \cap G|}{|C|}. \tag{9.4}$$

When $G = \emptyset$, we define $precision(C|\emptyset) = 1$. The two measures are adopted from information retrieval. Their uses in data mining have been considered by many authors [11, 17]. The inverse of precision is the error rate of G, that is, $error(C|G) = 1 - precision(C|G)$.

The probability used in the probabilistic model is, in fact, the precision of an equivalence class, namely, $Pr(C|[x]_Q) = precision(C|[x]_Q)$. In an (α,β)-approximation, the precision of an equivalence class in the positive region is larger than or equal to α, in the boundary region is between β and α, and in the negative region is smaller than or equal to β.

The following theorem gives conditions on the precision of the three regions.

Theorem 9.1. *In probabilistic (α,β)-approximations, the precisions of the three regions are bounded by:*

$$\begin{aligned} &precision(C|\mathrm{POS}_{Q,(\alpha,\beta)}(C)) \geq \alpha, \\ &\beta < precision(C|\mathrm{BND}_{Q,(\alpha,\beta)}(C)) < \alpha, \\ &precision(C|\mathrm{NEG}_{Q,(\alpha,\beta)}(C)) \leq \beta. \end{aligned} \tag{9.5}$$

Based on the connection between precision and recall, one can equivalently state these conditions in terms of an error rate. With the introduction of the two measures and the results of the above theorem, one can easily establish that the relationship between Pawlak $(1,0)$-approximations and probabilistic (α,β)-approximations:

$$\begin{aligned} &\alpha \leq precision(C|\mathrm{POS}_{Q,(\alpha,\beta)}(C)) \leq \\ &\quad precision(C|\mathrm{POS}_{Q,(1,0)}(C)) = 1, \\ &recall(C|\mathrm{POS}_{Q,(\alpha,\beta)}(C)) \geq \\ &\quad recall(C|\mathrm{POS}_{Q,(1,0)}(C)). \end{aligned} \tag{9.6}$$

The results clearly show the trade-off made between Pawlak and probabilistic models. For the positive region, a probabilistic model gains in recall, but loses in precision, and for the Pawlak, it is the reverse. Similar results can also be obtained for other regions.

Once the three regions are obtained, one may apply any rule induction algorithm to induce three-way rules.

Consider an interpretation of positive rules derived from the probabilistic positive region. Let $\mathrm{POS}_{Q,(\alpha,\beta)}(C)$ be the (α,β)-positive region of C. Suppose $H \subseteq \mathrm{POS}_{Q,(\alpha,\beta)}(C)$ and H is Q-definable, that is, H is a union of some Q-equivalence classes. Then, one can define a positive rule,

$$\mathrm{Des}(H) \rightarrow \mathrm{Des}(C),$$

where Des denotes a description of the set of objects H based on attribute-value pairs. An example for Table 9.1 is given by:

$$C = \{5,8,9,10\} = [(Flu, yes)],$$
$$\text{Des}(C) = (Flu, yes),$$
$$\text{POS}_{Q,(0.6,0.3)}(C) = \{3,4,5,7,8,9,10\};$$

$$H = \{7,8,9,10\} \subseteq \text{POS}_{Q,(0.6,0.3)}(C),$$
$$\text{Des}(H) = (Temperature, high);$$

$$r_1 : \ (Temperature, high) \rightarrow (Flu, yes);$$

$$precision(r_1) = precision(\{5,8,9,10\}|\{7,8,9,10\})$$
$$= 0.75 \geq 0.6.$$

In other words, for an Q-definable subset of $\text{POS}_{Q,(\alpha,\beta)}(C)$, we can induce a positive rule. The following theorem shows that the precision of such a rule is larger than or equal to α.

Theorem 9.2. *Suppose* $\text{Des}(H) \rightarrow \text{Des}(C)$ *is a positive rule in probabilistic* (α,β)*-approximations, that is,* H *is a non-empty and* Q*-definable subset of* $\text{POS}_{Q,(\alpha,\beta)}(C)$*. Then,* $precison(C|H) \geq \alpha$*.*

By this theorem, we may accept a decision with a precision larger than or equal to α, or equivalently with an error rate smaller than or equal to $1-\alpha$. For the special case of a Pawlak positive rule, the precision is 1, and the error rate is 0. Similar results can be stated for boundary and possible rules.

Detailed procedure of positive, boundary, and possible rules induction with LERS and the experimental evaluation are presented in the next three sections.

9.3 Positive and Possible Rule Induction with LERS

In this section, we introduce the LERS (Learning from Examples based on Rough Sets) data mining system and then discuss a method to transform a decision table so that LERS can be applied to induce types of rules, which are related to three-way rules.

The LERS data mining system computes lower and upper approximations for every concept and then induces rules using one of the selected algorithms [2, 3]. Rules induced from lower and upper approximations are called *certain* (or *positive*) and *possible* (or *plausible*), respectively [1]. Since the lower approximation is a subset of the upper approximation, certain rules may be viewed as a special case of possible rules.

A rule induction algorithm MLEM2 is one of the basic rule induction algorithms of LERS. MLEM2 explores the search space of attribute-value pairs. Its input data

set is a lower or upper approximation of a concept. In general, MLEM2 computes a *local covering* [2, 3] and then converts it into a rule set.

In the LERS format, every rule is associated with three numbers: the total number of attribute-value pairs on the left-hand side of the rule, the total number of cases correctly classified by the rule during training, and the total number of training cases matching the left-hand side of the rule, that is, the rule domain size.

The MLEM2 module of LERS induces the following set of certain rules:

1, 3, 3
$(Headache, no) \rightarrow (Flu, no)$,

and the following set of possible rules:

1, 3, 7
$(Headache, yes) \rightarrow (Flu, no)$,
1, 3, 3
$(Headache, no) \rightarrow (Flu, no)$,
1, 4, 7
$(Headache, yes) \rightarrow (Flu, yes)$.

9.4 Rules in Probabilistic Rough Sets

In order to induce probabilistic positive, boundary, and possible rules we have to modify input data sets. For every kind of rules (positive, boundary and possible) and a concept C described by $[(d, w)]$, the corresponding region will be unchanged (every entry will be the same as in the original data set). For all remaining cases, the decision value will be set to a special value, not listed in any attribute domain in the original data set, *for example*, let us use the value SPECIAL. Then, we will induce a possible rule set using the MLEM2 rule induction algorithm. From the induced rule set, only rules with (d, w) on the right-hand side will survive, all remaining rules (for other values of d and for values SPECIAL) should be deleted. The final rule set is a union of all rule sets computed this way separately for all values of d.

If we want to induce positive rules with $\alpha = 0.6$ and $\beta = 0.3$ for the data set presented on Table 9.1 we should construct two decision tables, presented as Tables 9.2 and 9.3.

Once those two regions are constructed, one can immediately adopt LERS to induce two sets of rules. However, two important issues remain to be investigated. One is the estimation of the parameters and the other is the interpretation of the induced rules.

From Table 9.2, the MLEM2 rule induction algorithm induced the following possible rule set:

1, 5, 6
$(Temperature, normal) \rightarrow (Flu, no)$,
2, 1, 3
$(Temperature, normal)$ & $(Headache, yes) \rightarrow (Flu, yes)$,
1, 4, 4
$(Temperature, high) \rightarrow (Flu, SPECIAL)$.

Table 9.2. A modified decision table to induce positive rules, with $\alpha = 0.6$ and $\beta = 0.3$, for (Flu, no)

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	normal	no	no	no
2	normal	no	no	no
3	normal	yes	no	no
4	normal	yes	no	no
5	normal	yes	no	yes
6	normal	no	yes	no
7	high	yes	yes	SPECIAL
8	high	yes	yes	SPECIAL
9	high	yes	yes	SPECIAL
10	high	yes	yes	SPECIAL

Table 9.3. A modified decision table to induce positive rules, with $\alpha = 0.6$ and $\beta = 0.3$, for (Flu, yes)

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	normal	no	no	SPECIAL
2	normal	no	no	SPECIAL
3	normal	yes	no	SPECIAL
4	normal	yes	no	SPECIAL
5	normal	yes	no	SPECIAL
6	normal	no	yes	SPECIAL
7	high	yes	yes	no
8	high	yes	yes	yes
9	high	yes	yes	yes
10	high	yes	yes	yes

From Table 9.3, the MLEM2 rule induction algorithm induced the following possible rule set:

1, 6, 6
$(Temperature, normal) \rightarrow (Flu, SPECIAL)$,
1, 1, 4
$(Temperature, high) \rightarrow (Flu, no)$,
1, 3, 4
$(Temperature, high) \rightarrow (Flu, yes)$.

The final set of probabilistic positive rules for Table 9.1 is as follows:

1, 5, 6
$(Temperature, normal) \rightarrow (Flu, no)$,
1, 3, 4
$(Temperature, high) \rightarrow (Flu, yes)$.

Similarly, for induction of boundary rules from Table 1, with $\alpha = 0.6$ and $\beta = 0.3$, we need to construct Table 9.4 for the only nonempty boundary region $\{3, 4, 5\}$.

Table 9.4. A modified decision table to induce probabilistic boundary rules, with $\alpha = 0.6$ and $\beta = 0.3$, for (Flu, yes)

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	normal	no	no	SPECIAL
2	normal	no	no	SPECIAL
3	normal	yes	no	no
4	normal	yes	no	no
5	normal	yes	no	yes
6	normal	no	yes	SPECIAL
7	high	yes	yes	SPECIAL
8	high	yes	yes	SPECIAL
9	high	yes	yes	SPECIAL
10	high	yes	yes	SPECIAL

From Table 9.4, the MLEM2 rule induction algorithm induced the following possible rule set:

1, 6, 6
$(Cough, yes) \rightarrow (Flu, SPECIAL)$,
1, 6, 6
$(Headache, no) \rightarrow (Flu, SPECIAL)$,
2, 2, 3
$(Temperature, normal)$ & $(Headache, yes) \rightarrow (Flu, no)$,
2, 1, 3
$(Temperature, normal)$ & $(Headache, yes) \rightarrow (Flu, yes)$.

For the concept [(*Flu*, *yes*)], the final set of probabilistic boundary rules with $\alpha = 0.6$ and $\beta = 0.3$ consists of the following single rule
2, 1, 3

$(Temperature, normal)$ & $(Headache, yes) \rightarrow (Flu, yes)$.

The set of boundary rules for the concept [(*Flu*, *no*)] is empty.

For the concept [(*Flu*, *no*)] and $\alpha = 0.6$ and $\beta = 0.3$, the probabilistic possible rule set is identical with the positive rule set. For the concept [(*Flu*, *yes*)] and $\alpha = 0.6$ and $\beta = 0.3$, the probabilistic possible rule set should be induced from the Table 9.5.

Table 9.5. A decision table

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	normal	no	no	SPECIAL
2	normal	no	no	SPECIAL
3	normal	yes	no	no
4	normal	yes	no	no
5	normal	yes	no	yes
6	normal	no	yes	SPECIAL
7	high	yes	yes	no
8	high	yes	yes	yes
9	high	yes	yes	yes
10	high	yes	yes	yes

1, 3, 3
$(Headache, no) \rightarrow (Flu, SPECIAL)$
1, 3, 7
$(Headache, yes) \rightarrow (Flu, no)$

1, 4, 7
$(Headache, yes) \rightarrow (Flu, yes)$

Thus, the set of probabilistic possible rules for the concept [(*Flu*, *yes*)] consists of the following single rule:

1, 4, 7
$(Headache, yes) \rightarrow (Flu, yes)$

9.5 Experiments

Our experiments were conducted using ten-fold cross validation on five data sets available on the University of California at Irvine *Machine Learning Repository*. For consistent data sets, for any values of alpha and beta, parameterized lower approximations are equal to upper ones, so it was necessary to use inconsistent data sets for our experiments. We converted consistent data sets into inconsistent ones by discretizing numerical attributes in such a way that the intervals, results of discretization, were too large. Data sets used for in our experiments are summarized in Table 9.6.

Table 9.6. Data sets used for experiments

Data set	Number of			Consistency
	cases	attributes	concepts	
Glass	214	9	6	55.14
Hepatitis	155	19	2	65.81
Postoperative patient	90	8	3	84.44
Primary tumor	339	17	21	72.27
Wine recognition	178	13	3	61.80

Obviously, parameterized lower approximations depend only on alpha, so for a fixed alpha parameters all corresponding lower approximations are equal to each other, hence the error rate, result of ten-fold cross validation is also the same. Similarly, for a fixed beta parameter, all corresponding upper approximations are identical, so corresponding error rates are also equal.

Results of the experiments are presented on Tables 9.7–9.11.

Table 9.7. Error rate for *glass* data set

	Type of rules:		
Parameters	positive	boundary	possible
$\alpha = 1.0, \beta = 0.0$	35.51	55.61	30.17
$\alpha = 1.0, \beta = 0.2$	35.51	58.41	31.78
$\alpha = 1.0, \beta = 0.4$	35.51	52.80	33.64
$\alpha = 1.0, \beta = 0.6$	35.51	59.35	32.24
$\alpha = 1.0, \beta = 0.8$	35.51	70.09	34.11
$\alpha = 0.8, \beta = 0.0$	34.11	60.28	30.17
$\alpha = 0.8, \beta = 0.2$	34.11	63.55	31.78
$\alpha = 0.8, \beta = 0.4$	34.11	60.28	33.64
$\alpha = 0.8, \beta = 0.6$	34.11	65.89	32.24
$\alpha = 0.6, \beta = 0.0$	32.24	69.63	30.17
$\alpha = 0.6, \beta = 0.2$	32.24	70.56	31.78
$\alpha = 0.6, \beta = 0.4$	32.24	66.36	33.64

Table 9.8. Error rate for *hepatitis* data set

	Type of rules:		
Parameters	positive	boundary	possible
$\alpha = 1.0, \beta = 0.0$	17.42	25.16	15.48
$\alpha = 1.0, \beta = 0.2$	17.42	23.87	15.48
$\alpha = 1.0, \beta = 0.4$	17.42	28.39	16.13
$\alpha = 1.0, \beta = 0.6$	17.42	20.65	15.48
$\alpha = 1.0, \beta = 0.8$	17.42	20.65	17.42
$\alpha = 0.8, \beta = 0.0$	17.42	76.13	15.48
$\alpha = 0.8, \beta = 0.2$	17.42	42.22	15.48
$\alpha = 0.8, \beta = 0.4$	17.42	50.97	16.13
$\alpha = 0.8, \beta = 0.6$	17.42	43.87	15.48
$\alpha = 0.6, \beta = 0.0$	15.48	75.48	15.48
$\alpha = 0.6, \beta = 0.2$	15.48	68.39	15.48
$\alpha = 0.6, \beta = 0.4$	15.48	79.35	16.13

Table 9.9. Error rate for *postoperative patient* data set

	Type of rules:		
Parameters	positive	boundary	possible
$\alpha = 1.0, \beta = 0.0$	37.78	55.56	42.22
$\alpha = 1.0, \beta = 0.2$	37.78	55.56	42.22
$\alpha = 1.0, \beta = 0.4$	37.78	53.33	38.89
$\alpha = 1.0, \beta = 0.6$	37.78	37.78	35.56
$\alpha = 1.0, \beta = 0.8$	37.78	—	37.78
$\alpha = 0.8, \beta = 0.0$	37.78	55.56	42.22
$\alpha = 0.8, \beta = 0.2$	37.78	55.56	42.22
$\alpha = 0.8, \beta = 0.4$	37.78	53.33	38.89
$\alpha = 0.8, \beta = 0.6$	37.78	37.78	35.56
$\alpha = 0.6, \beta = 0.0$	35.56	64.44	42.22
$\alpha = 0.6, \beta = 0.2$	35.56	64.40	42.22
$\alpha = 0.6, \beta = 0.4$	35.56	67.78	38.89

Table 9.10. Error rate for *primary tumor* data set

	Type of rules:		
Parameters	positive	boundary	possible
$\alpha = 1.0, \beta = 0.0$	69.32	79.94	63.32
$\alpha = 1.0, \beta = 0.2$	69.32	79.35	63.72
$\alpha = 1.0, \beta = 0.4$	69.32	80.83	62.24
$\alpha = 1.0, \beta = 0.6$	69.32	88.50	65.49
$\alpha = 1.0, \beta = 0.8$	69.32	79.94	69.07
$\alpha = 0.8, \beta = 0.0$	69.03	80.53	63.32
$\alpha = 0.8, \beta = 0.2$	69.03	79.94	63.72
$\alpha = 0.8, \beta = 0.4$	69.03	81.71	62.24
$\alpha = 0.8, \beta = 0.6$	69.03	89.68	65.49
$\alpha = 0.6, \beta = 0.0$	65.49	85.55	63.32
$\alpha = 0.6, \beta = 0.2$	65.49	85.55	63.72
$\alpha = 0.6, \beta = 0.4$	65.49	84.96	62.24

Table 9.11. Error rate for *wine recognition* data set

	Type of rules:		
Parameters	positive	boundary	possible
$\alpha = 1.0, \beta = 0.0$	37.08	47.19	12.36
$\alpha = 1.0, \beta = 0.2$	37.08	47.19	11.24
$\alpha = 1.0, \beta = 0.4$	37.08	47.75	10.11
$\alpha = 1.0, \beta = 0.6$	37.08	48.88	11.24
$\alpha = 1.0, \beta = 0.8$	37.08	50.56	13.48
$\alpha = 0.8, \beta = 0.0$	13.48	67.42	12.36
$\alpha = 0.8, \beta = 0.2$	13.48	69.66	11.24
$\alpha = 0.8, \beta = 0.4$	13.48	71.99	10.11
$\alpha = 0.8, \beta = 0.6$	13.48	71.91	11.24
$\alpha = 0.6, \beta = 0.0$	11.24	61.80	12.36
$\alpha = 0.6, \beta = 0.2$	11.24	66.29	11.24
$\alpha = 0.6, \beta = 0.4$	11.24	75.84	10.11

9.6 Conclusion

We conducted a number of experiments on five typical data sets. These data sets were preprocessed in order to increase inconsistency. Then, for different values of two parameters, alpha and beta, we evaluated positive, boundary, and possible rules, using ten-fold cross-validation.

The first, most obvious conclusion is that boundary rules are always the worst rules (the corresponding error rates are largest). It is caused by the fact that in induction of such rules the most important region, the positive region, is ignored. Due to this fact, the quality of boundary rules cannot exceed the quality of positive or possible rules.

As usually, it is difficult to tell whether positive or possible rules are better. For some data sets and some parameters, alpha and beta positive rules are better, for other data sets and other parameters, it is the other way around.

References

1. Grzymała-Busse, J.W.: Knowledge acquisition under uncertainty–A rough set approach. Journal of Intelligent & Robotic Systems 1, 3–16 (1988)
2. Grzymała-Busse, J.W.: LERS—a system for learning from examples based on rough sets. In: Słowiński, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)

3. Grzymała-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002), pp. 243–250. ESIA- Université de Savoie, France (2002)
4. Grzymała-Busse, J.W., Marepally, S.R., Yao, Y.: A comparison of positive, boundary, and possible rules using the MLEM2 rule induction algorithm. In: Proceedings of the 10th International Conference on Hybrid Intelligent Systems, pp. 7–12. IEEE Computer Society Press, Washington, DC (2010)
5. Grzymała-Busse, J.W., Yao, Y.: Probabilistic rule induction with the LERS data mining system. International Journal of Intelligent Systems 26, 518–539 (2011)
6. Grzymała-Busse, J.W., Ziarko, W.: Data mining based on rough sets. In: Wang, J. (ed.) Data Mining: Opportunities and Challenges, pp. 142–173. Idea Group Publ., Hershey (2003)
7. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
8. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
9. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29, 81–95 (1988)
10. Stefanowski, J.: Algorithms of Decision Rule Induction in Data Mining. Poznań University of Technology Press, Poznań (2001)
11. Tsumoto, S.: Accuracy and Coverage in Rough Set Rule Induction. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 373–380. Springer, Heidelberg (2002)
12. Tsumoto, S., Tanaka, H.: PRIMEROSE: probabilistic rule induction method based on rough sets and resampling methods. Computational Intelligence 11, 389–405 (1995)
13. Yao, Y.Y.: Decision-Theoretic Rough Set Models. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 1–12. Springer, Heidelberg (2007)
14. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180, 341–353 (2010)
15. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximate concepts. International Journal of Man-Machine Studies 37, 793–809 (1992)
16. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In: Raś, Z., Zemankova, M., Emrich, M. (eds.) Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems, Knoxville, Tennessee, October 25-27, pp. 388–395. Elsevier Press (1990)
17. Yao, Y.Y., Zhong, N.: An Analysis of Quantitative Measures Associated with Rules. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 479–488. Springer, Heidelberg (1999)
18. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)

Chapter 10
Exploring Neighborhood Structures with Neighborhood Rough Sets in Classification Learning

Qinghua Hu, Leijun Li, and Pengfei Zhu

Abstract. We introduce neighborhoods of samples to granulate the universe and use the neighborhood granules to approximate classification, thus they derived a model of neighborhood rough sets. Some machine learning algorithms, including boundary sample selection, feature selection and rule extraction, were developed based on the model.

Keywords: Classification learning, rough set, neighborhood, neighborhood entropy, sample selection, feature selection, attribute reduction, rule learning.

10.1 Introduction

Classification is a kind of important task in human cognition, reasoning and engineering applications. People learn to distinguish different classes of objects during all their life. Classification techniques also underlie numerous intelligent applications, including health diagnosis, fault analysis, object recognition, and so on. In these domains, a set of training samples are usually collected and each is described with a collection of features. Thus the samples are represented with feature vectors; the elements of the vectors are the feature values. In this case, the samples can be considered as sample points in the corresponding feature space. A learning algorithm is employed to extract classification rules from these data. The learned rules may be a continuous function or a set of discrete rules.

Qinghua Hu · Leijun Li · Pengfei Zhu
Harbin Institute of Technology
Box 458, Harbin 150001, Heilongjiang Province
P.R. China
e-mail: huqinghua@hit.edu.cn, lileijun1985@163.com, zhupengfeifly@gmail.com

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 277–307.
springerlink.com

Given a classification task, we have to ask a question how the different classes of objects can be distinguished from each other. As the objects of discourses are described with features, the objects with the same feature values are grouped into a subset and form an elemental concept labeled with a word, such as apple, banana, pear. Considering our cognition, we know there is a consistency assumption in classification modeling: the objects having the same feature values should be grouped into the same class; otherwise, the classification is inconsistent. The inconsistent objects that have the same feature values but belong to different classes form the boundary region of classification. Prof. Pawlak had an insight into this assumption and proposed the mathematical model of rough sets to compute the inconsistency due to the limited granularity of elemental concepts induced by finite knowledge [18, 19]. This assumption is widely used in our cognition and recognition. In the past, people have little knowledge about the world. We can just divide the objects in the world into coarse subsets. As we obtain more knowledge about the world, the objects are partitioned into finer subsets. Some objects that were not able to be precisely characterized in the past can be accurately described now. This phenomenon can currently be computed with the rough set model [19, 22]. From this viewpoint, the model of rough sets provides us an interesting and effective tool to understand human's cognition and extract knowledge from data.

In health diagnosis and engineering applications, objects are usually described with numerical features or mixed numerical and nominal variables. For example, temperature and blood pressure are widely gathered in medical analysis; the outputs of sensors in engineering applications are also numerical. In this context, small perturbation can be disregarded in classification. We take another consistency assumption in this case that the objects with similar feature values, instead of the same feature values, should be grouped into the same class; otherwise, the classification is not consistent. Here, the similar objects of object x can be considered as the neighborhood of x with respect to a certain similarity or dissimilarity function. If we assign a neighborhood to each sample, the family of neighborhoods generates a covering of the universe. We then utilize these neighborhood granules to approximately describe the subsets of the universe. The model of approximately describing classifications with neighborhood granules is called neighborhood rough sets.

In this work, we will introduce the basic concepts and properties of neighborhood rough sets; we also show the applications of neighborhood rough sets in classification. It is interesting to find that the boundary samples computed with neighborhood rough sets can be used to learn support vector machines. In addition, we construct attribute reduction algorithms based on the model. Unlike Pawlak's rough sets, this model does not require discretizing numerical features in attribute reduction. Finally, a rule extracting algorithm based on the neighborhood model is also described via neighborhood covering reduction. The related works have been reported in [9, 10, 8, 11, 12, 13].

10.2 Pawlak's Rough Sets and Related Works

In the early 1980s, Z. Pawlak introduced the model of rough sets, which is useful for machine learning and data mining, including feature selection, attribute reduction, rule extraction and uncertain reasoning [19].

Let $U = \{x_1, x_2, \cdots, x_n\}$ be a set of objects of discourse called the universe, and R be an equivalence relation on U. The ordered pair (U, R) is said to be an approximation space. $U/R = \{X_1, X_2, \cdots, X_k\}$ is the partition of U induced by R, where X_i is one of the equivalence classes. The objects in X_i are discernable as they are equivalent to each other with respect to R.

In applications, the objects in U are described with a set of features $A = \{a_1, a_2, \cdots, a_m\}$, and each object just takes a single value for a given feature. Given $B \subseteq A$, we associate an equivalence relation R over U that

$$R_B = \{(x_i, x_j) | \forall a \in B : a(x_i) = a(x_j), x_i, x_j \in U\}. \tag{10.1}$$

The equivalence class generated with $x_i \in U$ and B is denoted by $[x_i]_B$, which is a subset of objects with the same feature values as x_i with respect to B.

As to classification modeling, beside condition attributes A, we are also given a decision attribute D, which divides the objects into several decision classes $U/D = \{d_1, d_2, \cdots, d_N\}$. Take class d_i as an example. The lower and upper approximations of d_i in terms of $B \subseteq A$ are defined as

$$\underline{R_B} d_i = \{x | [x]_B \subseteq d_i, x \in U\}, \overline{R_B} d_i = \{x | [x]_B \cap d_i \neq \emptyset, x \in U\}. \tag{10.2}$$

considering the above definition, we can see that the lower approximation is a subset of samples whose equivalence classes consistently belong to the decision class d_i and the upper approximation is a subset of samples whose equivalence classes have elements in d_i. Obviously, $\overline{R_B} d_i \supseteq d_i \supseteq \underline{R_B} d_i$. The difference between $\overline{R_B} d_i$ and $\underline{R_B} d_i$ is called the boundary region, computed as $BND_B(d_i) = \overline{R_B} d_i - \underline{R_B} d_i$. $BND_B(d_i)$ is a subset of samples whose equivalence classes are not consistent. Some objects in it belong to d_i. However, there are also some objects belonging to other classes. Thus, it forms a region where samples cannot be certainly classified.

Correspondingly, we can also give the definition of lower and upper approximations of classification.

$$\underline{R_B} D = \bigcup_{i=1}^{N} \underline{R_B} d_i, \overline{R_B} D = \bigcup_{i=1}^{N} \overline{R_B} d_i. \tag{10.3}$$

The boundary region of classification is computed as $BND_B(D) = \overline{R_B} D - \underline{R_B} D$. It is easy to derive that $\overline{R_B} D = U$. So $BND_B(D) = U - \underline{R_B} D$. In fact, we can also obtain that $BND_B(D) = \bigcup_{i=1}^{N} BND_B(d_i)$.

As mentioned before, the decisions of samples in the boundary region are not consistent. They have the same feature values but belong to different decision classes, which leads to the confusion in classification. The size of boundary region reflects the capability of attribute set B in approximating D or the dependency of D on B. A function, called dependency, was defined as

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}. \tag{10.4}$$

$\gamma_B(D)$ reflects the complexity of classification using attribute set B. If $\gamma_B(D) = 1$, we say D is completely dependent on B. In this case, the classification is consistent. Every object can be precisely classified into a decision class according to its feature values. However, a lot of classification task is not consistent in real-world applications, then $\gamma_B(D) < 1$. Inconsistency maybe results from noise information or the limited knowledge. As to the first case two objects from different classes have the same feature values due to noise perturbation. As to the second case, the current information is not enough to discern objects from different classes. If new knowledge is introduced, the inconsistent samples may become distinguishable and the boundary region is reduced.

In data-driven classification modeling, the training samples with class labels are given to users. The users cannot discern these two classes of inconsistency in data. The task is to identify a classification function from these data. The learned function should have good generalization power on unseen objects (learning for prediction) or precisely describe the data (learning for description). It is notable that these two learning objectives sometimes lead to completely different learning strategies.

Rough sets based machine learning starts with attribute reduction, which provides a more compact representation of the raw task. A reduct is a sufficient and necessary subset B of features, satisfying the following conditions:

1) sufficient condition: $\gamma_B(D) = \gamma_A(D)$;
2) necessary condition: $\forall a \in B, \gamma_{B-a}(D) = \gamma_B(D)$.

As computing the optimal reduct is an NP-hard problem, a lot of heuristic attribute reduction algorithms have been developed these years [14, 20, 23, 25, 26, 27, 28, 31]. Moreover, a collection of rule extraction techniques were also developed [7, 24, 30].

10.3 Neighborhood Rough Sets

As mentioned before that objects are usually described with m numerical features. In this case, the samples can be considered as points in an m-dimensional feature space. The proximity or distance between samples generate a kind of structures of samples.

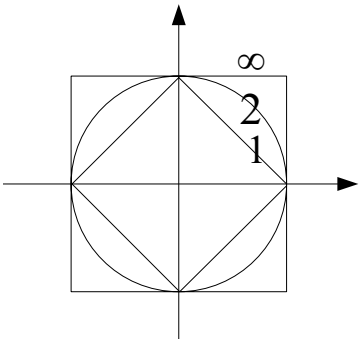


Fig. 10.1. Neighborhoods of x in terms of different distances

Definition 10.1. [10] Given $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of x_i in subspace B is defined as

$$\delta_B(x_i) = \{x_j | x_j \in U, \triangle_B(x_i, x_j) \leq \delta\}, \tag{10.5}$$

where Δ is a distance function. $\forall x_1, x_2, x_3 \in U$, it satisfies:

1) $\Delta(x_1, x_2) \geq 0, \Delta(x_1, x_2) = 0$ if and only if $x_1 = x_2$;
2) $\Delta(x_1, x_2) = \Delta(x_2, x_1)$;
3) $\Delta(x_1, x_3) \leq \Delta(x_1, x_2) + \Delta(x_2, x_3)$.

In applications, three distance functions are widely used. Consider x_1 and x_2 in an m-dimensional space $A = \{a_1, a_2, \cdots, a_m\}$, $v(x, a_i)$ denotes the value of sample x in the ith dimension a_i, then a general distance, named Minkowsky distance, is computed as

$$\Delta_P(x_1, x_2) = (\sum_{j=1}^{N} |v(x_1, a_i) - v(x_2, a_i)|^P)^{1/P}. \tag{10.6}$$

1) It is called Manhattan distance Δ_1 if $P = 1$; 2)Euclidean distance Δ_2 if $P = 2$; and 3) Chebychev distance if $P = \infty$. A detailed survey on distance functions can be found in some references.

$\delta_B(x_i)$ is a subset of samples which are close to sample x_i . The size of neighborhood depends on the threshold δ, which can be viewed as the parameter of granularity. More samples fall into $\delta_B(x_i)$ if δ increases. The shape of the neighborhood is determined by the used distance function. Considering the 2-dimensional real space, neighborhoods of x_0 in terms of the above three metrics are as shown in Figure 10.1. 1-norm-based neighborhood is a rhombus region around center sample ; 2-norm based neighborhood is a ball region, while the infinite-norm-based neighborhood is a rectangle or a square.

Besides the above functions, there also are some distance functions for mixed numerical and categorical data [9], such as Heterogeneous Euclidean-Overlap Metric function (HEOM), Value Difference Metric (VDM), Heterogeneous Value

Difference Metric (HVDM) and Interpolated Value Difference Metric (IVDM). The HEOM is defined in the following way

$$HEMO(x,y) = \sqrt{\sum_{j=1}^{m} w_{a_i} \times d_{a_i}^2(x_{a_i}, y_{a_i})}, \tag{10.7}$$

where m is the number of attributes, w_{a_i} is the weight of attribute a_i, $d_{a_i}(x,y)$ is the distance between samples x and y in terms of attribute a_i being defined as

$$d_{a_i}(x,y) = \begin{cases} 1, & if\ the\ attribute\ value \\ & of\ x\ or\ y\ is\ unknown, \\ overlap_a(x,y), & if\ a\ is\ a\ nominal\ attribute, \\ rn\_diff_a(x,y), & if\ a\ is\ a\ numerical\ attribute. \end{cases} \tag{10.8}$$

Here, $overlap_a(x,y) = \begin{cases} 0, & if\ x = y \\ 1, & othwise \end{cases}$ and $rn\_diff_a(x,y) = \frac{|x-y|}{max_a - min_a}$.

With different metric functions, the proposed technique can be used to analyze categorical attributes, numerical attributes, interval-valued attributes, and their mixtures.

If confronted with objects described with other types of features, such as time series, images, and graphs, we can also compute the neighborhood of the objects using some corresponding distance or similarity functions, such as dynamic time warping distance for time series, histogram intersection kernel for image classification.

From the above discussion, we can see there are two key factors in definition of neighborhood. One is the used distance function, and the other is threshold δ. The first one determines the shape of neighborhoods and the latter controls the size of neighborhoods. Furthermore, a neighborhood granule degrades to an equivalence class if $\delta = 0$. In this case, the samples in the same neighborhood are equivalent and the neighborhood rough set model degrades to the classical rough set model.

In order to deal with heterogeneous features, we here compute the neighborhoods of samples in the following way [9].

Definition 10.2. Let $B_1 \subseteq A$ and $B_2 \subseteq A$ be numerical attributes and categorical attributes, respectively. The neighborhood granules induced by B_1, B_2, and $B_1 \bigcup B_2$ are defined as

(1) $\delta_{B_1}(x) = \{x_i | \Delta_{B_1}(x,x_i) \leq \delta, x_i \in U\}$;

(2) $\delta_{B_2}(x) = \{x_i | \Delta_{B_2}(x,x_i) = 0, x_i \in U\}$;

(3) $\delta_{B_1 \bigcup B_2}(x) = \{x_i | \Delta_{B_1}(x,x_i) \leq \delta \wedge \Delta_{B_2}(x,x_i) = 0, x_i \in U\}$; where $\wedge$ means "and" operator.

The first item is designed for the objects described with numerical attributes, the second is for categorical attributes, and the last one is for numerical and categorical attributes. Thus this definition is able to deal with data with numerical, categorical and their mixed features.

In computing neighborhoods, if the numerical features are normalized into the unit interval [0, 1] and the values of categorical features are coded as a series of integral numbers, such as $1,2,3\cdots$, distance between two objects is no less than 1 if they take different values in any categorical feature. Generally speaking, δ is specified a value far smaller than 1. In this case, the objects with different categorical values necessarily do not belong to the same neighborhood. This suggests that two objects belong to the same neighborhood only if they take the same values in all categorical features.

Given a metric space $< U, \Delta >$, the collection of neighborhood granules $\{\delta(x_i)|x_i \in U\}$ forms an elemental granule system, which covers the universe, rather than partitions. We have

1) $\forall x_i \in U, \delta(x_i) \neq \emptyset$;
2) $\bigcup_{x \in U} \delta(x) = U$.

A neighborhood relation $\mathcal{N}$ on the universe can be written as a relation matrix $M(\mathcal{N}) = (r_{ij})_{n \times n}$ where

$$r_{ij} = \begin{cases} 1, & \Delta(x_i, x_j) \leq \delta \\ 0, & otherwise. \end{cases} \tag{10.9}$$

It is easy to show that $\mathcal{N}$ satisfies the following properties: reflexivity: $r_{ii} = 1$; symmetry: $r_{ij} = r_{ji}$. Obviously, neighborhood relations are a class of similarity relations, which satisfy reflexivity and symmetry. Neighborhood relations draw the objects together for similarity or indistinguishability in terms of the distances between objects.

Remark 1. $\delta(x)$ is an equivalence class and $\mathcal{N}$ is an equivalence relation if we specify $\delta = 0$. This is applicable to categorical data.

Remark 2. According to Definition 10.2, if numerical and categorical features coexist, the samples within a neighborhood granule are close to each other in terms of the numerical features; in the mean time, they are equivalent with respect to the categorical attributes.

Definition 10.3. [10] Giving a set of samples U, $\mathcal{N}$ is a neighborhood relation on U, $\{\delta(x_i)|x_i \in U\}$ is the family of neighborhood granules. Then we call $< U, \mathcal{N} >$ a neighborhood approximation space.

Definition 10.4. [10] Given $< U, \mathcal{N} >$, for arbitrary $X \subseteq U$, two subsets of objects, called lower and upper approximations of X in terms of relation $\mathcal{N}N$, are defined as

$$\underline{\mathcal{N}}X = \{x_i|\delta(x_i) \subseteq X, x_i \in U\}, \tag{10.10}$$

$$\overline{\mathcal{N}}X = \{x_i|\delta(x_i) \cap X \neq \emptyset, x_i \in U\}. \tag{10.11}$$

Accordingly, the boundary region of X in the approximation space is formulated as

$$BN(X) = \overline{\mathcal{N}}X - \underline{\mathcal{N}}X. \tag{10.12}$$

Table 10.1. Play tennis data with heterogeneous features

Day	*outlook*	*T*	*humidity*	*windy*	*play*
x1	sunny	85	85	false	no
x2	sunny	80	90	true	no
x3	overcast	83	86	false	yes
x4	rainy	70	96	false	yes
x5	rainy	68	80	false	yes
x6	rainy	65	70	true	no
x7	overcast	64	65	true	yes
x8	sunny	72	95	false	no
x9	sunny	69	70	false	yes
x10	rainy	75	80	false	yes
x11	sunny	75	70	true	yes
x12	overcast	72	90	true	yes
x13	overcast	81	75	false	yes
x14	rainy	71	91	true	no

Example 1. Table 10.1 gives a data set play tennis with mixed categorical and numerical features, where $U = \{x1, x2, \cdots, x14\}$, A={outlook, T, humidity, windy}, D={play}, and condition attributes outlook and windy are categorical; T and humidity are numerical and play is the decision.

As for classical rough sets, features T and humidity should be discretized before rough set analysis is performed. Here we directly analyze it with neighborhood model.

As an example, we discuss features *outlook* and T. According Definition 10.2, attribute outlook groups the 14 samples into 3 subsets:

U/outlook = $\{X_1 = \{x1, x2, x8, x9, x11\}, X_2 = \{x3, x7, x12, x13\}, X_3 = \{x4, x5, x6, x10, x14\}\}$.

Here $X_1 \bigcup X_2 \bigcup X_3 = U$ and $X_i \cap X_j = \emptyset$ if $i \neq j$.

As to numerical feature T, we specify $\delta = 5$, then

$|x2 - x1| \leq 5$, so $x2 \in \delta(x1)$. Similarly, we get $\delta(x1) = \{x1, x2, x3, x13\}, \delta(x2) = \{x1, x2, x3, x10, x11, x13\}, \cdots, \delta(x14) = \{x4, x5, x8, x9, x10, x11, x12, x14\}$.

It is easy to see $\bigcup_{i=1}^{14} \delta(x_i) = U$ and $\delta(x1) \cap \delta(x2) = \{x1, x2, x3, x13\} \neq \emptyset$.

With these two features, the samples in $\delta(x1)$ should take value sunny with attribute outlook, and the distance with $x1$ should be not greater than 5 with T. $\delta_{outlook,T}(x1) = \{x1, x2\}$. Similarly, $\delta_{outlook,T}(x2) = \{x1, x2, x11\}, \delta_{outlook,T}(x3) = \{x3, x13\}, \cdots, \delta_{outlook,T}(x14) = \{x4, x5, x14\}$. Assumed $X = \{x1, x2, x6, x8, x14\}$, namely, the samples with decision no, $\delta_{outlook,T}(x1) \subseteq X$, so $x1 \in \underline{\mathcal{N}X}$, similarly, we can get $\underline{\mathcal{N}X} = \{x1\}, \overline{\mathcal{N}X} = \{x1, x2, x4, x5, x6, x8, x9, x10, x11, x14\}$.

The size of boundary region reflects the degree of roughness of set X in the approximation space. Assuming X is the sample subset with a decision label, usually

we hope the boundary region of the decision could be as little as possible for decreasing uncertainty in decision. The sizes of boundary regions depend on X, attributes to describe U, and threshold δ. Delta is the parameter to control analysis granularity, and we have the following theorem.

Theorem 10.1. *[9] Given* $< U, \Delta, \mathcal{N} >$ *and two nonnegative* δ_1 *and* δ_2, *if* $\delta_1 \geq \delta_2, \mathcal{N}_1 \supseteq \mathcal{N}_2$, *we have*

1)$\forall x_i \in U : \delta_1(x_i) \supseteq \delta_2(x_i)$;

2) $\forall X \subseteq U : \underline{\mathcal{N}_1} X \subseteq \underline{\mathcal{N}_2} X, \overline{\mathcal{N}_1} X \subseteq \overline{\mathcal{N}_2} X$,

where $\mathcal{N}_1$ *and* $\mathcal{N}_2$ *are the neighborhood relations induced with* δ_1 *and* δ_2, *respectively.*

Proof. $\delta_1 \geq \delta_2$, we have $\delta_1(x_i) \supseteq \delta_2(x_i)$. Assuming $\delta_1(x_i) \subseteq X$, we have $\delta_2(x_i) \subseteq X$. Therefore we must have $x_i \in \underline{\mathcal{N}_2} X$ if $x_i \in \underline{\mathcal{N}_1} X$. However, $x_i$is not sure in $\underline{\mathcal{N}_1} X$ if we have $x_i \in \underline{\mathcal{N}_2} X$. Hence $\underline{\mathcal{N}_1} X \subseteq \underline{\mathcal{N}_2} X$. Similarly, we can get $\overline{\mathcal{N}_1} X \subseteq \overline{\mathcal{N}_2} X$.

The above theorem shows that a finer neighborhood relation is produced with a smaller delta; accordingly, the lower approximation is larger than that with a large delta. In neighborhood rough sets, parameter delta provides a chance to explore the data under different granularity.

Definition 10.5. [10] Given a neighborhood decision table $NDT =< U, A \bigcup D >$, $X_1, X_2, \cdots, X_N$ are the classes with decisions 1 to N, $\mathcal{N}_B$ is the neighborhood information granule generated by attributes $B \subseteq A$, the lower and upper approximations of the decision D with respect to attribute set B are defined as

$$\underline{\mathcal{N}_B} D = \bigcup_{i=1}^{N} \underline{\mathcal{N}_B} X_i, \quad \overline{\mathcal{N}_B} X = \bigcup_{i=1}^{N} \overline{\mathcal{N}_B} X_i, \tag{10.13}$$

where

$$\underline{\mathcal{N}_B} X = \{x_i | \delta(x_i) \subseteq X, x_i \in U\}, \overline{\mathcal{N}_B} X = \{x_i | \delta(x_i) \cap X \neq \emptyset, x_i \in U\}. \tag{10.14}$$

The decision boundary region of D with respect to attributes B is defined as

$$BN(D) = \overline{\mathcal{N}_B} X - \underline{\mathcal{N}_B} D. \tag{10.15}$$

The lower approximation of decision, also called positive region of decision, denoted by $POS_B(D)$, is the subset of objects whose neighborhood subset consistently belongs to one of the decision classes, while the neighborhood subsets of the boundary samples come from more than one decision class. As to classification learning, these samples are one of the sources causing classification complexity because they confuse the employed learning algorithm. They take the similar or even equivalent feature values but belong to different decision classes.

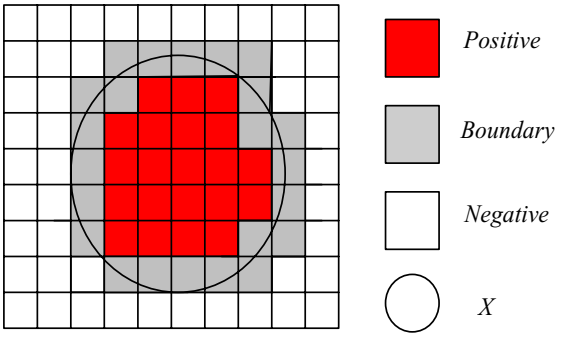


Fig. 10.2. Rough set in discrete feature space

It is easy to show that

1) $\overline{\mathcal{N}_B}D = U$;
2) $POS_B(D) \cap BN(D) = \emptyset$;
3) $POS_B(D) \bigcup BN(D) = U$.

Therefore, the neighborhood model divides the samples into two disjoint groups: positive region and boundary region. Positive region is the sample set which can be classified into one of the classes without uncertainty according to the existing attributes, while boundary region is the set of samples which cannot be determinately classified. Intuitively, the samples in boundary region are easy to be misclassified. In data acquirement and preprocessing, one usually tries to find a feature space in which the classification task has the least boundary.

Example 2. Approximations are demonstrated as shown in Figures 10.2 and 10.3. In the discrete case, the samples are granulated into a number of mutually exclusive equivalence information granules with their feature values, shown as the lattices in Figure 10.2. Assuming we want to describe a subset $X \subseteq U$ with these granules, then we will find two subsets of granules: a maximal subset of granules which are included in X and a minimal subset of granules which includes X. Figure 10.3 shows an example of binary classification in a 2-D numerical space, where d_1 is labeled with "$*$" and d_2 is labeled with "$+$". Taking samples x_1, x_2, and x_3 as examples, we assign spherical neighborhoods to these samples. We can find $\delta(x_1) \subseteq d_1$ and $\delta(x_3) \subseteq d_2$, while $\delta(x_2) \cap d_1 \neq \emptyset$ and $\delta(x_2) \cap d_2 \neq \emptyset$. According to the above definitions, $x_1 \in \underline{\mathcal{N}} d_1, x_3 \in \underline{\mathcal{N}} d_2$, and $x_2 \in BN(D)$. As a whole, regions A_1 and A_3 are decision positive regions of d_1 and d_2, respectively, while A_2 is the boundary region.

The size of a boundary region reflects the degree of roughness of set X in the approximation space. Assuming X is the sample subset with a decision label, usually we hope the boundary region of the decision could be as little as possible for decreasing uncertainty in decision. The sizes of boundary regions depend on X, the attributes used to describe U, and threshold δ.

The samples in different feature subspaces will have different boundary regions. The size of boundary region reflects the discernibility of the classification problem in

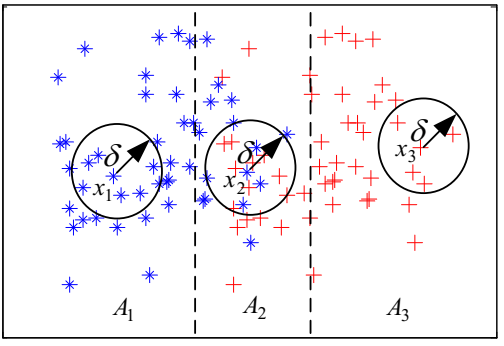


Fig. 10.3. Rough set in numerical feature space

Fig. 10.4. Two binary classification tasks

the corresponding subspaces. It also reflects the recognition power or characterizing power of the condition attributes. The greater the boundary region is, the weaker the characterizing power of the condition attributes is. It can be formulated as follows.

Definition 10.6. The dependency degree of D to B is defined as the ratio of the objects in the positive region:

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}. \tag{10.16}$$

$\gamma_B(D)$ reflects the ability of B to approximate D. Obviously, $0 \leq \gamma_B(D) \leq 1$. We say D completely depends on B if $\gamma_B(D) = 1$, denoted by $B \Rightarrow D$; otherwise we say $D\gamma - depends$ on B, denoted by $B \Rightarrow_\gamma D$. If $\gamma_B(D) = 1$, we say the classification D is consistent in terms of attribute set B and granularity δ.

Example 3. Geometrical interpretation of dependence: As shown in Figure 10.4(1), if the patterns are complete classifiable, then, the boundary sample set is empty. Hence, $\gamma_B(D) = |POS_B(D)|/|U| = |A \bigcup B|/|A \bigcup B| = 1$. In this case, the decision is completely dependent on attribute subset B. However, if there is an overlapped region between classes, as shown in Figure 10.4(2), that is, there are some inconsistent samples. The dependency is computed as:

$$\gamma_B(D) = |POS_B(D)|/|U| = |A \bigcup C|/|A \bigcup B \bigcup C| < 1.$$

We can find that the dependency function depends on the size of the intersection between classes. Generally speaking, we hope to find a feature subspace, in which, the classification problem is with the least overlapping region. If the samples are completely separable (linearly separable or nonlinearly separable), the dependency is 1. we say the classification is consistent; otherwise, we say it is inconsistent. With an inconsistent classification problem, we try to find the feature subset which gets the largest dependence.

Theorem 10.2. *[9] (Attribute Monotonicity). Given a neighborhood decision system* $< U, A \bigcup D >, B_1, B_2 \subseteq A, B_1 \subseteq B_2$*, with the same metric in computing neighborhoods, we have*

1) $\mathcal{N}_{B1} \supseteq \mathcal{N}_{B2}$*;*

2) $POS_{B_1}(D) \subseteq POS_{B_2}(D)$*,*

2) $\forall X \subseteq U, \underline{\mathcal{N}_{B1}} X \subseteq \underline{\mathcal{N}_{B2}} X; \gamma_{B_1}(D) \leq \gamma_{B_2}(D)$.

Proof. With the same metric $\forall x \in U$, we have $\delta_{B_1}(x) \supseteq \delta_{B_2}(x)$ if $B_1 \subseteq B_2$. Assume $\delta_{B_1}(x) \subseteq \underline{\mathcal{N}_{B_1}} X$, where X is one of the decision classes, then we have $\delta_{B_2}(x) \subseteq \underline{\mathcal{N}_{B_2}} X$. In the same time, there may be $x_i, \delta_{B_1}(x_i) \not\subset \underline{\mathcal{N}_{B_1}} X$, and $\delta_{B_2}(x) \subseteq \underline{\mathcal{N}_{B_2}} X$. Therefore, $POS_{B_1}(D) \subseteq POS_{B_2}(D)$. Accordingly, we have $\gamma_{B_1}(D) \leq \gamma_{B_2}(D)$.

Theorem 10.2 shows dependence monotonously increases with attributes, which means that adding a new attribute in the attribute subset at least does not decrease the dependence. This property is very important for constructing feature selection algorithms. Generally speaking, we hope to find a minimal feature subset which has the same characterizing power as the whole samples. The monotony of the dependency function is very important for constructing a greedy forward or backward search algorithm[8]. It guarantees that adding any new feature into the existing subset does not lead a decrease in the significance of the new subset.

Definition 10.7. Given a neighborhood decision table $NDT =< U, A \bigcup D, V, f >, B \subseteq A$, we say attribute subset B is a relative reduct if

1) Sufficiency: $\gamma_B(D) = \gamma_A(D)$;

2) Necessity: $\forall a \in B, \gamma_B(D) > \gamma_{B-a}(D)$.

The first condition guarantees that $POS_B(D) = POS_A(D)$. The second condition shows there is no superfluous attribute in the reduct. Therefore, a reduct is the minimal subset of attributes which has the same approximating power as the whole attribute set.

10.4 Information Entropy for Neighborhood Models

The relationship between Shannon's information entropy and Z. Pawlak's rough sets was well discussed [4, 28]. However, Shannon's entropy is not applicable to numerical information and the mutual information derived in this model cannot be used

to compute relevance between numerical features. In this section, we introduce the concept of neighborhood into information theory, and generalize Shannon's entropy to neighborhood information entropy [11].

Definition 10.8. Given a set of samples $U = \{x_1, x_2, \cdots, x_n\}$ described by numerical or discrete features A. $B \subseteq A$ is a subset of attributes. The neighborhood of sample x_i in B is denoted by $\delta_B(x_i)$, then the neighborhood uncertainty of the sample x_i is defined as

$$NH_\delta^{x_i}(B) = -\log \frac{\|\delta_B(x_i)\|}{n}, \tag{10.17}$$

and the average uncertainty of the set, also called neighborhood entropy, of samples is computed as

$$NH_\delta(B) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_B(x_i)\|}{n}. \tag{10.18}$$

Since $\forall x_i, \delta_B(x_i) \subseteq U, \|\delta_B(x_i)\|/n \leq 1$, so we have $\log n \geq NH_\delta(B) \geq 0. NH_\delta(B) = \log n$ if and only if for $\forall x_i, \|\delta_B(x_i)\| = 1. NH_\delta(B) = 0$ if and only if for $\forall x_i, \|\delta_B(x_i)\| = n$.

Theorem 10.3. *Assume that* $\delta \leq \delta', NH_\delta(B) \geq NH'_\delta(B)$.

Proof. $\forall x_i \in U$, we have $\delta(x_i) \subseteq delta'(x_i)$, then $\|\delta(x_i)\| \leq \|\delta'(x_i)\|$, we have $NH_\delta(B) \geq NH'_\delta(B)$.

Theorem 10.4. $NH_\delta(B) = H(B)$ *if* $\delta = 0$*, where* $H(B)$ *is Shannon's entropy.*

Proof. If $\delta = 0$, the samples are partitioned into disjoint $X_1, X_2, \cdots, X_m$, where $\Delta(x_i, x_j) = 0$ if $x_i, x_j \in X_k$. Assume there are m_i samples in X_i. Then $H(B) = -\sum_{i=1}^{m} \frac{m_i}{n} \log \frac{m_i}{n}$. $\delta_B(x) = X_k$ if $x \in X_k$ and $\delta = 0$. If $i \neq j, X_i \cap X_j = \emptyset$, we have

$$NH_\delta(B) = -\frac{1}{n}\sum_{i} \log \frac{\|\delta_B(x_i)\|}{n} = \sum_{x \in X_1} -\frac{1}{n} \log \frac{\|\delta_B(x_i)\|}{n} + \cdots + \sum_{x \in X_m} -\frac{1}{n} \log \frac{\|\delta_B(x_i)\|}{n}$$

So $NH_\delta(B) = -\sum_{i=1}^{m} \frac{m_i}{n} \log \frac{m_i}{n}$. This leads us to the conclusion that $NH_\delta(B) = H(B)$.

Neighborhood entropy is a natural generalization of the Shannon's entropy if features are continuous. As to discrete features, we can define a discrete distance such that $\Delta(x, y) = 0$ if $x = y$; otherwise $\Delta(x, y) = 1$. If $\Delta < 1$, the subset $\delta_B(x_i)$ of samples forms the equivalence class $[x_i]$, where $[x_i]$ is the set of samples taking the same feature values with x_i. In this case, the neighborhood entropy equals Shannon entropy.

Definition 10.9. $B,C \subseteq A$ are two subsets of attributes. The neighborhood of sample x_i in feature subspace $B \bigcup C$ is denoted by $\delta_{B \bigcup C}(x_i)$, then the joint neighborhood entropy is computed as

$$NH_\delta(B,C) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_{B \bigcup C}(x_i)\|}{n} \tag{10.19}$$

Especially if B is a set of input variables, C is the classification attribute, we define $delta_{B \bigcup C}(x_i) = \delta_B(x_i) \cap c_{x_i}$. Thus

$$NH_\delta(B,C) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_R(x_i) \cap c_{x_i}\|}{n}$$

Theorem 10.5. $NH_\delta(B,C) \geq NH_\delta(B), NH_\delta(B,C) \geq NH_\delta(C)$.

Proof. $\forall x_i \in U$, we have $\delta_{B \bigcup C}(x_i) \subseteq \delta_B(x_i)$ and $\delta_{B \bigcup C}(x_i) \subseteq \delta_C(x_i)$. Then, $\|\delta_{B \bigcup C}(x_i)\| \leq \|\delta_B(x_i)\|$ and $\|\delta_{B \bigcup C}(x_i)\| \leq \|\delta_C(x_i)\|$; therefore, $NH_\delta(B,C) \geq NH_\delta(B), and\ NH_\delta(B,C) \geq NH_\delta(C)$.

Definition 10.10. $B,C \subseteq A$ are two subsets of attributes. The conditional neighborhood entropy of B to C is defined as

$$NH_\delta(B|C) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_{B \bigcup C}(x_i)\|}{\|\delta_C(x_i)\|} \tag{10.20}$$

Theorem 10.6. $NH_\delta(B|C) = NH_\delta(B,C) - NH_\delta(C)$

Proof. $NH_\delta(B,C) - NH_\delta(C)$

$$\begin{aligned}
&= -\tfrac{1}{n}\sum_{i=1}^{n} \log \tfrac{\|\delta_{B \bigcup C}(x_i)\|}{n} - (-\tfrac{1}{n}\sum_{i=1}^{n} \log \tfrac{\|\delta_C(x_i)\|}{n}) \\
&= -\tfrac{1}{n}\sum_{i=1}^{n} (\log \tfrac{\|\delta_{B \bigcup C}(x_i)\|}{n} - \log \tfrac{\|\delta_C(x_i)\|}{n}) \\
&= -\tfrac{1}{n}\sum_{i=1}^{n} \log \tfrac{\|\delta_{B \bigcup C}(x_i)\|}{\|\delta_C(x_i)\|}
\end{aligned}$$

Definition 10.11. $B,C \subseteq A$ are two subsets of attributes. The neighborhood mutual information of B and C is defined as

$$NMI_\delta(B;C) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_B(x_i)\| \cdot \|\delta_C(x_i)\|}{n\|\delta_{B \bigcup C}(x_i)\|} \tag{10.21}$$

Theorem 10.7. *Given two subsets of attributes B and C, $NMI_\delta(B;C)$ is the mutual information of these subsets, then the following equations hold:*

1) $NMI_\delta(B;C) = NMI_\delta(C;B)$;
2) $NMI_\delta(B;C) = NH_\delta(B) + NH_\delta(C) - NH_\delta(B,C)$;
3) $NMI_\delta(B;C) = NH_\delta(B) - NH_\delta(B|C) = NH_\delta(C) - NH_\delta(C|B)$.

Proof. The conclusions of 1) and 3) are straightforward; here we give the proof of property 2).

$$
\begin{aligned}
&2) NH_\delta(B) + NH_\delta(C) - NH_\delta(B,C) \\
&= -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_B(x_i)\|}{n} - \frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_C(x_i)\|}{n} - (-\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_{B\cup C}(x_i)\|}{n}) \\
&= -\frac{1}{n}\sum_{i=1}^{n} (\log \frac{\|\delta_B(x_i)\|}{n} + \log \frac{\|\delta_C(x_i)\|}{n} - \log \frac{\|\delta_{B\cup C}(x_i)\|}{n}) \\
&= -\frac{1}{n}\sum_{i=1}^{n} (\log \frac{\|\delta_B(x_i)\|}{n} \times frac\|\delta_C(x_i)\|n \times \frac{n}{\|\delta_{B\cup C}(x_i)\|}) \\
&= -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\|\delta_B(x_i)\|\cdot\|\delta_C(x_i)\|}{n\|\delta_{B\cup C}(x_i)\|}
\end{aligned}
$$

Lemma 10.1. *Given a set U of samples described by attribute set $A, B \subseteq A$ and D is the decision attribute. $NMI_\delta^x(B;D) = H^x(B)$ if the decision of sample $x \in U$ is δ-neighborhood consistent, where $NMI_\delta^x(B;D) = -\log \frac{\|\delta_B(x)\|\cdot\|D_x\|}{n\|\delta_{B\cup D}(x)\|}, H^x(D) = -\log \frac{\|D_x\|}{n}$.*

Proof. $\delta_{B\cup D}(x) = \delta_B(x) \cap D_x$, and we have that $\delta_B(x) \subseteq D_x$ if x is consistent. So $delta_{B\cup D}(x) = \delta_B(x)$. Then $-\log \frac{\|\delta_B(x)\|\cdot\|D_x\|}{n\|\delta_{B\cup D}(x)\|} = -\log \frac{\|\delta_B(x)\|\cdot\|D_x\|}{n\|\delta_B(x)\|} = -\log \frac{\|D_x\|}{n}$.

Theorem 10.8. *Given a set of samples U described by the attribute set $A, B \subseteq A$ and D is the decision attribute. $NMI_\delta(B;D) = H(D)$ if the samples in feature subspace B are δ-neighborhood consistent.*

Proof. As the decisions of samples in feature subspace are consistent, the decision of each sample is consistent. For $\forall x_i \in U, NMI_\delta^{x_i}(B;D) = H^{x_i}(D)$. So $\sum_{i=1}^{n} NMI_\delta^{x_i}(B;D) = NMI_\delta(B;D)$; $\sum_{i=1}^{n} H^{x_i}(D) = H(D)$. We get the conclusion that $NMI_\delta(B;D) = H(D)$.

The above theorem shows that the mutual information between features B and decision D is equal to the uncertainty quantity of decision if the classification is consistent with respect to the knowledge of B. There is not any uncertainty in classification if attributes B is known. Moreover, we also know by The lemma that the mutual information between B and D with respect to sample x is the uncertainty of x in classification if its decision is consistent. With Lemma 10.1 and Theorem 10.6, we not only distinguish whether all samples in classification learning are consistent, but also know which samples are consistent if there are some inconsistent samples in classification. In practice it is often that just some, instead of all, samples are consistent. It is useful to find these consistent patterns for understanding the classification task at hand.

10.5 Boundary Sample Selection with Neighborhood Model

The samples near the classification border usually have larger influence on the classification model than other samples. In addition, these samples can help users

understand the classification tasks. For example, the border samples are difficult to be distinguished. If we know which samples are located near the border, we can extract additional features to discern them.

In[2], Cortes and Vapnik showed that the weights of optimal classification hyperplane can be written as linear combination of support vectors, which are located near the border. One can select only a part of the samples, so-called support vectors, to train SVM, rather than the whole training sets if they can be found in advance. By this way, the learning time and space complexity will be greatly reduced. Based on this observation, some researches were reported to select patterns for SVM [15, 16, 21]. Shin and Cho discussed a neighborhood property based pattern selection algorithm, where neighborhood entropy was defined. They associated each samples with k nearest neighbors, then checked the entropy of the neighborhood. If the entropy is not zero, then the samples are considered as boundary set [21]. Furthermore, they gave the proof that neighborhood relation between training samples in input space is preserved in feature space. This lays a firm groundwork for boundary sample selection.

In neighborhood rough sets, the difference of the lower and upper approximations is called boundary region. Obviously, the samples in the boundary region are near the samples from different classes. These samples are possibly support vectors. If we use neighborhood rough sets to compute these samples and train support vector machines just with them. The learning processing would significantly speed up.

In neighborhood rough sets, $BN(D) = \overline{\mathcal{N}_B}D - \underline{\mathcal{N}_B}D = U - \underline{\mathcal{N}_B}D$. If we get the positive samples, then we can compute the boundary samples. According to the definition of boundary samples, we know if the neighborhood of a sample is not consistent, this sample belongs to classification boundary region. Consider sample x, we can check whether there is a sample within x's neighborhood having different class label. If yes, we collect x in the boundary set.

Look at Figure 10.5(1). We can see there is a boundary region in the binary classification task. In addition, there are two noisy samples. We search the boundary set with neighborhood rough sets at granularity δ_1. We can see in Figure 10.5(2) that x_1 and x_2 are far from the classification border. They are class-noisy samples. In classification learning, we should omit these samples; otherwise, the model may be confused. Here the idea of variable precision rough sets can be combined with neighborhood rough sets to deal with this problem [29].

Definition 10.12. Given two sets X and Y , we define the inclusion degree of X in Y as

$$I(X,Y) = |X \cap Y|/|X| \tag{10.22}$$

Definition 10.13. [12] Given a family of neighborhood information granules $\delta(x_i)$, $i = 1,2,\cdots,n, X \subseteq U$, the variable precision lower approximation and upper approximation is defined as

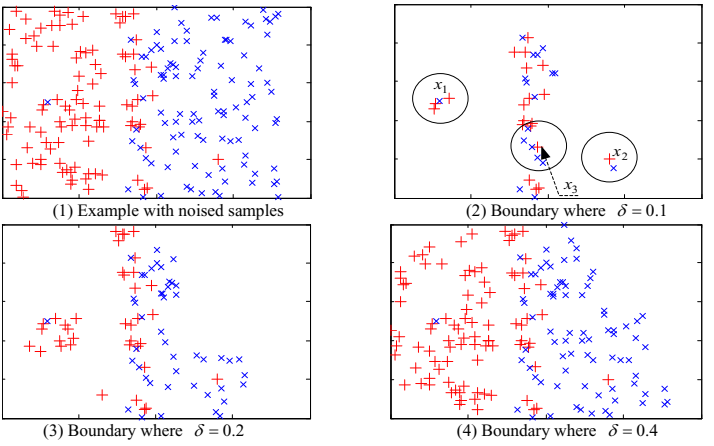


Fig. 10.5. An example data with noised samples

$$\underline{\mathcal{N}^{\beta}}X = \{x_i | I(\delta_B(x_i), X) \geq \beta, x_i \in U\}, \tag{10.23}$$

$$\overline{\mathcal{N}^{\beta}}X = \{x_i | I(\delta_B(x_i), X) > 1 - \beta, x_i \in U\}, \tag{10.24}$$

where $\beta > 0.5$.

Definition 10.13 relaxes the condition of strict inclusion or strict exclusion in Definition 4 and replaces them with majority inclusion and majority exclusion. Based on the variable precision neighborhood model, x_1, x_2 and their neighborhood will be classified into the positive regions of class 1 and class 2, respectively. While samples like x_3 are still grouped into the decision boundary.

Figures 10.5(2), (3), and (4) present the boundary samples discovered with neighborhood rough set model, where $\delta = 0.1, \delta = 0.2, \delta = 0.4$, respectively. Obviously the boundary region increases with threshold δ. And we also find some of the samples far away decision boundary also be included in the boundary set due to the noised samples. These samples will have negative effect on the classification hyperplane. They should be deleted from the boundary set.

Here we redefine the positive region and decision boundary as follows:

$$POS_B(D) = \{x_i | \exists d_j, I(\delta(x_i), d_j) \geq \beta\}, \tag{10.25}$$

$$BN(D) = \{x_i | \forall d_j, I(\delta(x_i), d_j) < \beta\}, \tag{10.26}$$

We also have

$$U = POS_B(D) \bigcup BN(D). \tag{10.27}$$

In this case, the noise in positive region can be recognized as

$$N(D) = \{x_i | x_i \in d_k, \exists d_j, I(\delta(x_i), d_j) \geq \beta, k \neq j\}. \tag{10.28}$$

By specifying a proper threshold δ and β, we can find a boundary region with appropriate size and delete the noisy samples from the boundary set. Several works discussed the problem of specifying the value of parameter β. At large, β can take values in arrange [0.7, 1]. The value of δ depends on applications. Generally speaking, if the inter-class distance of a learning sample set is large, we should assign δ with a large value to get enough boundary samples to support the optimal hyperplane, vice versa. Generally, δ can take value in arrange [0.1, 0.5] if numerical attributes are normalized to the unit interval [0, 1].

Figure 10.6 shows some toy examples. There are three typical classification problems. The first is a binary classification problem with circle classification plane. The second is a 4×4 checkerboard classification problem, and the third one is a three-class problem. Figures 10.6(1-1), (2-1) and (3-1) give the raw sample set; Figures 10.6(2-1), (2-2) and (3-2) present the optimal classification planes trained with the raw data; Figures 10.6(1-3), (2-3) and (3-3) show the boundary samples found with 1-norm neighborhood rough set model. Finally, Figures 10.6(1-4), (2-4) and (3-4) show the optimal hyperplanes trained just with the boundary samples computed with neighborhood rough sets. We can see that the two classes of classification planes are quite similar to each other although most of the learning samples don't take part in the training process after sample reduction.

10.6 Feature Selection with Neighborhood Model

The dependence function reflects the approximating capability of a condition attribute set in approximating decisions. It can be used to measure the quality of the attributes. The objective of attribute reduction is to search a subset of attributes such that the classification problem has the maximal dependency; meanwhile, no redundant attribute exists in the selected subset. In this section, we construct some measures for attribute selection and reduction and then present a greedy feature selection algorithm.

Definition 10.14. Given a neighborhood decision table $< U, A \bigcup D >, B \subseteq A, \forall a \in B$, one can define the significance of a in B as

$$Sig_1(a, B, D) = \gamma_B(D) - \gamma_{B-a}(D). \tag{10.29}$$

Note that an attribute's significance is the function of three variables: a, B and D. An attribute a may be of great significance in B_1 but of little significance in B_2. What's more, the attribute's significance is different for each decision if they are multiple decision attributes in a decision table.

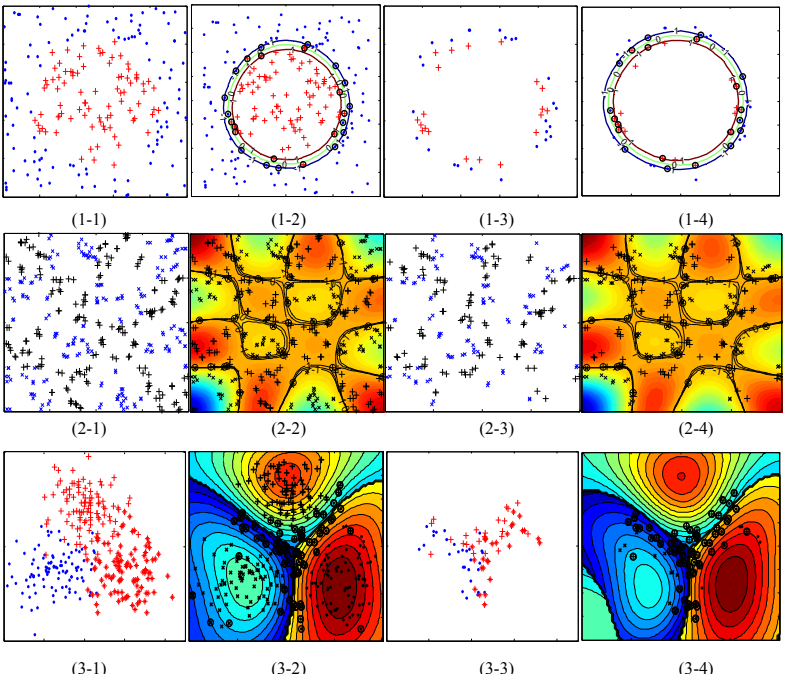


Fig. 10.6. Three toy examples

The above definition is applicable to backward feature selection. Similarly, a measure applicable to forward selection can be defined as shown below.

Definition 10.15. Given a neighborhood decision table $< U, A \bigcup D >, B \subseteq A, \forall a \in A - B$, one can define the significance of a in B as

$$Sig_2(a,B,D) = \gamma_{B \bigcup a}(D) - \gamma_B(D). \tag{10.30}$$

As $0 \leq \gamma_B(D) \leq 1$ and $\forall a \in B : \gamma_B(D) \geq \gamma_{B-a}(D)$, we have

$$0 \leq Sig_1(a,B,D) \leq 1, 0 \leq Sig_2(a,B,D) \leq 1 \tag{10.31}$$

We say attribute a is superfluous in B with respect to D if $Sig_1(a,B,D) = 0$; otherwise, a is indispensable.

The objective of rough set based attribute reduction is to find a subset of attributes which has the same approximating power as the original data and has no redundant attribute. With the above measures, forward greedy search algorithms for attribute reduction can be formulated as follows [9].

Algorithm 1 : **Naive forward attribute selection based on neighborhood rough sets** (**NFARNRS**)

Input : $< U, A \bigcup D >$

Delta // Control the size of the neighborhood Output: reduct *red*.

1: $\emptyset \rightarrow red$; // *red* is the pool to contain the selected attributes

2: For each $a_i \in A - red$

3: Compute $\gamma_{red \bigcup a_i}(D) = \frac{|POS_{B \bigcup a_i}(D)|}{|U|}$

4: Compute $SIG(a_i, red, D) = \gamma_{red \bigcup a_i}(D) - \gamma_{red}(D)$

5: end

6: select the attribute a_k satisfying $SIG(a_k, red, D) = \max_i(SIG(a_i, red, D))$

7: If $SIG(a_k, red, D) > \varepsilon$, // ε is a little positive real number use to control the convergence

8: $red \bigcup a_k \rightarrow red$

9: go to step2

10: else

11: return *red*

12: end if

There is a parameter of delta, which is d to control the size of neighborhoods, to be specified in the algorithm. Usually, parameter delta takes value in the interval [0.1, 0.3] if all numerical features are normalized into [0, 1].

There are some key steps in a feature selection algorithm: subset generation, subset evaluation and stopping criterion. In algorithm NFAS-NRS, we begin with an empty set red of attribute and add one feature which makes the increment of dependency maximal into the set red in each round. We evaluate the subset with dependency function. The attributes maximizing the increment of dependency are selected. The algorithm stops until the dependency does not increase by adding any new feature into the attribute subset red.

According to the Geometrical interpretation of dependence function, we see that the algorithm tries to find the feature subspace such that there is the least overlapped region between classes for a given classification task. We achieve the goal by maximizing the positive region, accordingly, maximizing the dependency between the decision and condition attributes. The samples in the boundary region are easy to be misclassified. Intuitively, the accuracy of the classification increases as the number of boundary samples decreases.

NFAS-NRS computes the relation between each pair of samples in every round. The worst case of computational complexity is $O(m^2n^2)$, where m and n are the numbers of features and samples, respectively. Assumed there are k attributes included in the reduct, the total computation times are

$$m \times n^2 + (m-1) \times n^2 + \cdots + (m-k) \times n^2 = (2m-k)(k+1) \times n^2/2. \quad (10.32)$$

As we know the positive region of decision is monotonous with the attributes, we have the following corollary which can be used to speed up the algorithm.

Property 10.1. Given a neighborhood decision system $< U, A \bigcup D, f >$ and a metric $\Delta, M, N \subseteq A, M \subseteq N$, if $x_i \in POS_M(D)$ then $x_i \in POS_N(D)$.

Property 10.1 shows that an object necessarily belongs to the positive region with respect to an attribute set if it belongs to the positive region with respect to its subset. In the forward attribute selection, the attributes are added into the selected subset one by one according to their significances, correspondingly, according the size of positive regions. $\forall M \supseteq B, x_i \in POS_M(D)$, if object $x_i \in POS_B(D)$. Therefore, we need not compute the objects in $POS_B(D)$ when compute $POS_M(D)$ because they are necessarily in $POS_M(D)$. In this case, we just need consider the objects in $U - POS_B(D)$. The objects in $U - POS_B(D)$ and the rested features get fewer and fewer as the attribute selection goes on, so the computation will be reduced in selecting a new feature. Based on this observation, a fast forward algorithm is formulated as follows [9].

Algorithm 2 : **Fast attribute selection based on neighborhood rough sets** (**FAS** − **NRS**)
Input : $< U, A \bigcup D >$ delta // Control the size of the neighborhood
Output : reduct *red*.
1:$\emptyset \rightarrow red, U \rightarrow S$; // is used to contain the selected attributes, S is the set of samples out of positive region.
2: while $S \neq \emptyset$
3: for each $a_i \in A - red$
4: generate a temporary decision table $DT_i =< U, red \bigcup a_i, D >$
5: $\emptyset \rightarrow POS_i$
6: for each $O_j \in S$
7: Compute $\delta(O_j)$ in neighborhood decision table DT_i
8: if $\exists X_k \in U/D$, such that $\delta(O_j) \subseteq X_k$
9: $POS_i \bigcup O_j \rightarrow POS_i$
10: end if
11: end for
12: end for
13: find a_k such that $|POS_k| = \max_i |POS_i|$
14: if $POS_k \neq \emptyset$
15: $red \bigcup a_k \rightarrow red$
16: $S - POS_k \rightarrow S$
17: else
18: exit while
19: end if
20: end while
21: return *red*, end

Assume there are n samples in the decision table and k features in the reduct, and selecting an attribute averagely adds n/k samples into the positive region, the computational times of reduction are

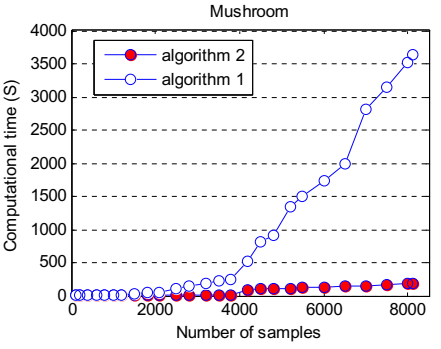


Fig. 10.7. Comparison of computational efficiency (mushroom)

$$N \times n \times n + (N-1) \times n \times \frac{k-1}{k} \times n + \cdots + (N-k) \times \frac{1}{k} \times n \times n < \frac{n^2}{k} \times (k + k - 1 + \cdots + 1) \tag{10.33}$$

Thus the computational complexity is $N \times n^2(k+1)/2$. In practice, it is usually found that most of samples have been grouped into positive regions if a few features have been selected. Therefore the computation will be greatly reduced at the sequential round and the reduction procedure speeds up.

In order to test the effectiveness of FAS-NRS, we collect two data sets, mushroom and letters. Mushroom is a data set including 8124 samples described with 22 features, and letters including 20000 samples with 16 features. We randomly draw some samples and conduct attribute selection on the drawn data sets. The drawn data gradually increase to the whole sets. We observe the change of computational cost, as shown in Figures 10.7 and 10.8. We compute the distance between different samples and find their neighborhood subsets. It is easy to see that the computational time and comparison times of algorithm 1 quickly increase with the numbers of samples; however, the computation complexity of Algorithms 2 is not so sensitive to the size of sample sets compared with Algorithm 1. With 8124 samples from mushroom, Algorithm 3 conducts 581188 comparisons in reduction, whereas Algorithm 1 requires 21665485 comparisons. About 97% computations are reduced. The similar case occurs to data letter.

10.7 Rule Extraction with Neighborhood Model

Extracting rules from training samples is a kind of important tasks in machine learning and classification modeling. Covering reduction is a classical technique to extract rules. As we know, neighborhoods of samples form a covering of the universe. In this work, we introduce a new way to specify values of parameter delta, which are calculated according to the location of samples in feature spaces. So the values of delta of samples may be different. Here we set as the classification margin of samples [6].

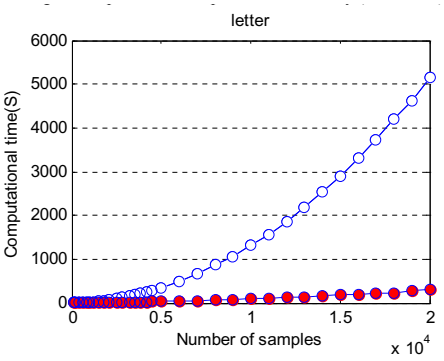


Fig. 10.8. Comparison of computational efficiency (letter)

However, in Definition 10.1, all samples are specified with the same value of delta. In order to differentiate them, here we give a new definition of neighborhood.

Definition 10.16. [13] Given arbitrary $x_i \in U$, the δ neighborhood $\mathcal{N}(x_i)$ in feature space is defined as

$$\mathcal{N}(x_i) = \{x_j \in U : \Delta(x_i, x_j) \leq \delta\}, \tag{10.34}$$

where Δ is a distance function and δ is a parameter dependent on x_i.

The key difference of this definition is that the size of neighborhood varies with sample points. Now we give an approach to computing δ.

Given a set of samples $< U, A, D >, x \in U. NH(x)$ is the nearest sample of x within the same class of x, called the nearest hit of x, while $NM(x)$ is the nearest sample of x outside of the class of x, called the nearest miss of x. Then the classification margin of x is computed as

$$m(x) = \Delta(x, NM(x)) - \Delta(x, NH(x)) \tag{10.35}$$

The margin of a sample x reflects how much the features of x can be corrupted by noise before x is misclassified. If $m(x) < 0$, the sample will be misclassified if we use the nearest neighbor rule because x is closer to the samples in other classes. In this case we set $M(x) = 0$, and $\mathcal{N}(x_i) = \{x_j : \Delta(x_i, x_j) = 0\}$; otherwise, the value of δ of sample x is set as $M(x)$ when we compute the neighborhood of x.

Certainly, the size of neighborhood can be specified in other ways, which leads to different neighborhood coverings.

The family of neighborhoods $\mathcal{N} = \{\mathcal{N}(x_1), \mathcal{N}(x_2), \cdots, \mathcal{N}(x_n)\}$ generates a covering of the universe. Now we call $< U, \mathcal{N}, D >$ a neighborhood covering decision system.

Definition 10.17. [13] Let $< U, \mathcal{N}, D >$ be a neighborhood covering decision system. $X \subseteq U$ is an arbitrary subset of U. The lower and upper approximations of X in $< U, \mathcal{N}, D >$ are defined as

$$\underline{\mathcal{N}_1}X = \{x \in U : \mathcal{N}(x) \subseteq X\}, \underline{\mathcal{N}_2}X = \{\mathcal{N}(x) : \mathcal{N}(x) \subseteq X\}; \quad (10.36)$$

$$\overline{\mathcal{N}_1}X = \{x \in U : \mathcal{N}(x) \cap X \neq \emptyset\}, \overline{\mathcal{N}_2}X = \{\mathcal{N}(x) : \mathcal{N}(x) \cap X \neq \emptyset\}. \quad (10.37)$$

where $\underline{\mathcal{N}_1}X$ and $\underline{\mathcal{N}_2}X$ are called Type 1 and Type 2 covering lower approximations, respectively, and $\overline{\mathcal{N}_1}X$ and $\overline{\mathcal{N}_2}X$ are Type 1 and Type 2 covering upper approximations, respectively.

If the neighborhood covering $\mathcal{N}$ is also a partition of the universe, then $\underline{\mathcal{N}_1}X = \underline{\mathcal{N}_2}X$ and $\overline{\mathcal{N}_1}X = \overline{\mathcal{N}_2}X$. However, usually $\mathcal{N}$ is not a partition. In this case the two definitions are different.

Example 1. $U = \{x_1, x_2, x_3, x_4, x_5\}, \mathcal{N}(x_1) = \{x_1, x_2\}, \mathcal{N}(x_2) = \{x_1, x_2, x_3\}$, $\mathcal{N}(x_3) = \{x_2, x_3, x_4\}, \mathcal{N}(x_4) = \{x_3, x_4, x_5\}, \mathcal{N}(x_5) = \{x_4, x_5\}$. Assume that $X = \{x_1, x_2, x_3\}$. We compute $\underline{\mathcal{N}_1}X = \{x_1, x_2\}, \underline{\mathcal{N}_2}X = \{x_1, x_2, x_3\}, \overline{\mathcal{N}_1}X = \{x_1, x_2, x_3, x_4\}$, $\overline{\mathcal{N}_2}X = \{x_1, x_2, x_3, x_4, x_5\}$.

Definition 10.18. [13] Let $< U, \mathcal{N}, D >$ be a neighborhood covering decision system. $U/D = \{X_1, X_2, \cdots, X_l\}$is the partition of U induced with D. The lower and upper approximations of decision D are defined as

$$\underline{\mathcal{N}_1}D = \bigcup_{i=1}^{l} \underline{\mathcal{N}_1}X_i, \underline{\mathcal{N}_2}D = \bigcup_{i=1}^{l} \underline{\mathcal{N}_2}X_i; \quad (10.38)$$

$$\overline{\mathcal{N}_1}D = \bigcup_{i=1}^{l} \overline{\mathcal{N}_1}X_i, \overline{\mathcal{N}_2}D = \bigcup_{i=1}^{l} \overline{\mathcal{N}_2}X_i. \quad (10.39)$$

It is easy to see that $\overline{\mathcal{N}_1}D = U$ and $\overline{\mathcal{N}_2}D = U$. However, $\underline{\mathcal{N}_1}D \subseteq U$ and $\underline{\mathcal{N}_2}D \subseteq U$.

Definition 10.19. [13] Let $< U, \mathcal{N}, D >$ be a neighborhood covering decision system. $U/D = \{X_1, X_2, \cdots, X_l\}, x \in U$, we say x is Type-1 consistent if there exists $X_i \in U/D$, such that $\mathcal{N}(x) \subseteq X_i$. In this case $\underline{\mathcal{N}_1}D = U$. We say x is Type-2 consistent if there exists $x^{'} \in U$ and $X_i \in U/D$, such that $x \in \mathcal{N}(x^{'})$and $\mathcal{N}(x^{'}) \subseteq X_i$. In this case, $\underline{\mathcal{N}_2}D = U$.

Property 10.2. If $\underline{\mathcal{N}_1}D = U$, then $\underline{\mathcal{N}_2}D = U$ holds. However, if $\underline{\mathcal{N}_2}D = U$, we cannot obtain $\underline{\mathcal{N}_1}D = U$.

$\underline{\mathcal{N}_1}D = U$ shows that the neighborhood of each sample is consistent. Thus, there exists $X_i \in U/D$, such that $\mathcal{N}(x) \subseteq X_i$. Therefore, $\underline{\mathcal{N}_2}D = U$. Whereas when $\underline{\mathcal{N}_2}D = U$, there may be some inconsistent covering element $\mathcal{N}(x)$. In this case, $\underline{\mathcal{N}_1}D \neq U$.

Definition 10.20. If all the samples in U are Type-1 consistent, namely $\underline{\mathcal{N}_1}D = U$, we say that the neighborhood covering decision system is Type-1 consistent; otherwise we say the system is Type-1 inconsistent. If all the samples in U are Type-2 consistent, namely $\underline{\mathcal{N}_2}D = U$, we say that the neighborhood covering decision system is Type-2 consistent; otherwise we say the system is Type-2 inconsistent.

Example 2. $U = \{x_1,x_2,x_3,x_4,x_5\}, \mathcal{N}(x_1) = \{x_1,x_2\}, \mathcal{N}(x_2) = \{x_1,x_2,x_3\}, \mathcal{N}(x_3) = \{x_2,x_3,x_4\}, \mathcal{N}(x_4) = \{x_3,x_4,x_5\}, \mathcal{N}(x_5) = \{x_4,x_5\}$. Assume that $U/D = \{X_1,X_2\}, X_1 = \{x_1,x_2,x_3\}$ and $X_2 = \{x_4,x_5\}$. Then we get $\underline{\mathcal{N}_1}X_1 = \{x_1,x_2\}, \underline{\mathcal{N}_1}X_2 = \{x_5\}, \underline{\mathcal{N}_2}X_1 = \{x_1,x_2,x_3\}, \underline{\mathcal{N}_2}X_2 = \{x_4,x_5\}$ So $\underline{\mathcal{N}_1}D = \{x_1,x_2,x_5\}$, while $\underline{\mathcal{N}_1}D = \{x_1,x_2,x_3,x_4,x_5\}$. The decision system is Type-2 consistent and Type-1 inconsistent.

There maybe exist two classes of covering elements in Type 2 consistent neighborhood covering decision systems: consistent and inconsistent covering elements. $\mathcal{N}(x_1), \mathcal{N}(x_2)$ and $\mathcal{N}(x_5)$ are consistent, while $\mathcal{N}(x_3)$ and $\mathcal{N}(x_4)$ are inconsistent. However, all the covering elements are consistent if the decision system is Type-1 consistent.

It is also reasonable that inconsistent covering elements exist in consistent covering decision systems. As to a covering decision system, there are some redundant covering elements. Although we can find a neighborhood granule for each sample such that this sample is consistent in this granule, this sample may also exist in other inconsistent neighborhood granules. We just require finding a granule which contains this sample and the granule is consistent.

Definition 10.21. Let $< U, \mathcal{N}, D >$ be a Type-1 or Type-2 consistent neighborhood covering decision system. X_i is one of the decision classes. $\mathcal{N}(x^{'}) \in \mathcal{N}$. If $\exists \mathcal{N}(x) \in \mathcal{N}$, such that $\mathcal{N}(x^{'}) \subseteq \mathcal{N}(x) \subseteq X_i$, we say $\mathcal{N}(x^{'})$ is relatively consistent reducible with respect to X_i; otherwise, we say $\mathcal{N}(x^{'})$ is relatively consistent irreducible.

Definition 10.22. [13] Let $< U, \mathcal{N}, D >$ be a type-2 consistent neighborhood covering decision system. If $\mathcal{N}(x) \in \mathcal{N}$ is an inconsistent covering element, we say $\mathcal{N}(x)$ is a relatively inconsistent reducible element.

There are two types of reducible elements: one is consistent and is contained by other consistent elements; the other is the inconsistent elements.

Definition 10.23. [13] Let $< U, \mathcal{N}, D >$ be a type-1 consistent neighborhood covering decision system. If $\forall \mathcal{N}(x) \in \mathcal{N}$, there does not exist $\mathcal{N}(x^{'}) \in \mathcal{N}$, such that $\mathcal{N}(x^{'}) \subseteq \mathcal{N}(x) \subseteq X_i$, where X_i is an arbitrary decision class, then we say $< U, \mathcal{N}, D >$ is relatively irreducible; otherwise, we say $\mathcal{N}(x^{'})$ is relatively reducible.

Definition 10.24. [13] Let $< U, \mathcal{N}, D >$ be a type-1 consistent neighborhood covering decision system. $\mathcal{N}^{'} \subseteq \mathcal{N}$ is a derived covering from $\mathcal{N}$ by reducing the redundant covering elements, and $< U, \mathcal{N}^{'}, D >$ is relatively irreducible. Then we say that $\mathcal{N}^{'}$ is a D-relative reduct of $\mathcal{N}$, denoted by $reduct_D(\mathcal{N})$.

Property 10.3. Let $< U, \mathcal{N}, D >$ be a type-1 consistent neighborhood covering decision system and $reduct_D(\mathcal{N})$ be a D-relative reduct of $\mathcal{N}$. Then $< U, reduct_D(\mathcal{N}), D >$ is also a type-1 consistent covering decision system, and $\forall \mathcal{N}(x) \in \mathcal{N}, \exists \mathcal{N}(x^{'}) \in reduct_D(\mathcal{N})$, such that $\mathcal{N}(x) \subseteq \mathcal{N}(x^{'})$.

The conclusions of Property 10.3 are straightforward because we just remove the redundant covering elements in the covering. Moreover, as to a type-1 consistent covering decision system, all the elements are consistent; naturally the reduced covering decision system is also type-1 consistent.

After covering element reduction, there is no redundant covering element in the covering decision system. All the selected covering elements are useful in approximating the decision classes. With a reduct of a covering decision system, we can generate covering rules in the form

If $x^{'} \in \mathcal{N}(x)$, then $x^{'}$ is assigned with the class of $\mathcal{N}(x)$.

The theoretic framework of neighborhood covering reduction forms a mechanism for classification rule learning from training samples. In next section, we will construct an algorithm for rule learning based on neighborhood covering reduction.

We now introduce a novel rule learning algorithm based on neighborhood covering reduction. Just like the problem of minimal attribute reduction, the search of minimal rule set is also NP-hard [1]. There are several strategies to search the suboptimal rule set, such as forward search, backward search and genetic algorithm [17]. Here we consider the forward search technique which start with an empty set of rules, and add new rules one by one. In each step, the consistent neighborhood which covers most samples is selected and generates a piece of rule.

In addition, we can see that some neighborhoods just cover several samples in the experiments. If we include these rules, the size of rule base is very large and the corresponding classification model would overfit the training samples. Thus a pruning strategy is required. The pruning techniques used in other rule learning systems are applicable [5]. In this work, we add rules one by one [13]. In the meanwhile, we also test the current set of rules with training and test samples. We record the classification accuracies. The rule set yielding the best classification performance is outputted. Certainly, other pruning techniques can also adopted.

Algorithm NCR : **Rule Learning Based on Neighborhood Covering Reduction**

Input : Training set: $U_{train} = \{(x_1, d_1), \cdots, (x_i, d_i), \cdots, (x_n, d_n)\}, i = 1, 2, \cdots, n$;
Test set $U_{test} = \{(x_1^{'}, d_1^{'}), \cdots, (x_j^{'}, d_j^{'}), \cdots, (x_m^{'}, d_m^{'})\}, j = 1, 2, \cdots, m$.

Output : rule set $R = \{r_1, \cdots, r_i, \cdots, r_h\}$, where the rule is in the form of $(x_i, m(x_i), d - i)$.

1: compute the margin $m(x_i)$ of each sample $x_i, i = 1, 2, \cdots, n$. If $m(x_i) < 0$, we set $m(x_i) = 0$.
2: compute the neighborhood covering element $\mathcal{N}(x_i)$ of sample $x_i, i = 1, 2, \cdots, n$, the covering of the universe is denoted by $\mathcal{N}$.
3. Count the samples covered by each covering element in $\mathcal{N}$.
4: $R \leftarrow \emptyset$
5: While $(\mathcal{N} \neq \emptyset)$
6: select the cover element $\mathcal{N}(x)$ which covers the most samples.
7: add a rule $(x, m(x), y)$ into the rule set R, where $m(x)$ is the margin of x, y is the decision of x.

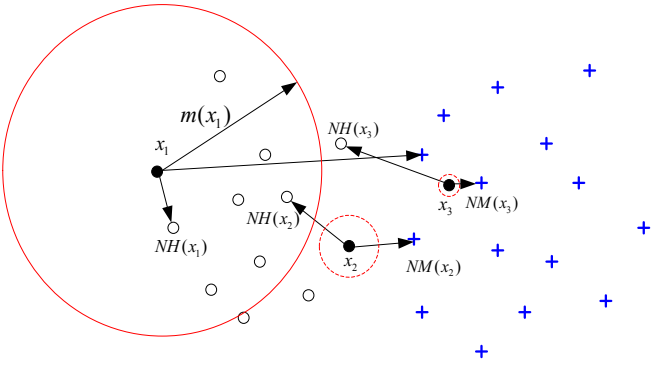


Fig. 10.9. Demonstration of neighborhood covering elements

8: remove the samples which are covered by $\mathcal{N}(x)$.
9: compute the classification of rule set R on the training or test samples
10: end
11: sort the rules according to the size of the covering elements in the descending order
12: choose the first h rules that produce the highest classification accuracy on the training sample set U_train or U_test

This algorithm greedily searches the largest neighborhood of samples in the forward search step and removes the rules covering the least samples. In this way, we generate a small set of rules which can cover most of the samples.

It is worth pointing out that the above algorithm is developed according to type 2 neighborhood rough sets, instead of type 1 neighborhood rough sets. However, if the size of neighborhood is set as the margin of samples, there is not any inconsistent neighborhood that derived neighborhood covering decision system. So the rules produced with the above algorithm are consistent.

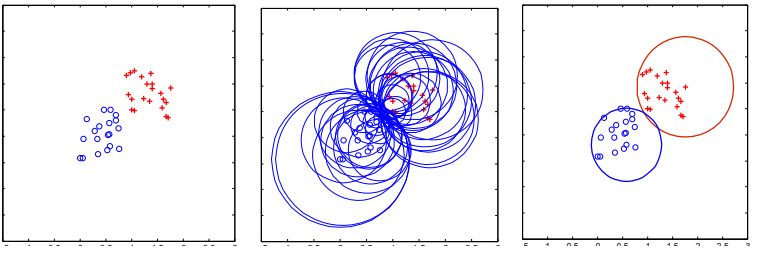
According to the definition of margin of samples, we know the margins of samples far away from the classification border are large. So their neighborhoods are also larger than the neighborhoods of samples close to classification border, as shown in Figure 10.1. There are two classes of samples in this classification task, where "$\circ$" is a sample from d_1 , and "$+$" from d_2.

Now we consider samples x_1, x_2 and x_3.

$$m(x_1) = \Delta(x_1, nm(x_1)) - \Delta(x_1, nh(x_1)) \geq 0;$$
$$m(x_2) = \Delta(x_2, nm(x_2)) - \Delta(x_2, nh(x_2)) \approx 0;$$
$$m(x_3) = \Delta(x_3, nm(x_3)) - \Delta(x_3, nh(x_3)) < 0.$$

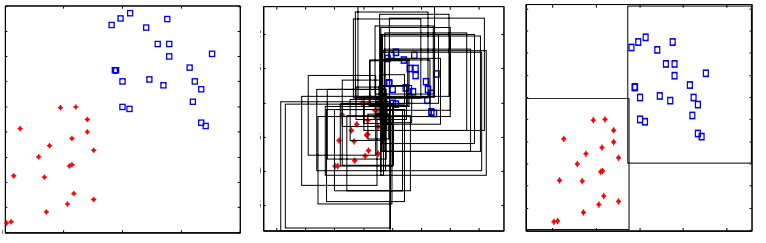
Correspondingly, $\mathcal{N}(x_2) = \{x_2\}$ and $\mathcal{N}(x_3) = \{x_3\}$. They are pruned in the pruning step, whereas $\mathcal{N}(x_1)$ covers many samples and it is selected and generates a classification rule that if $\Delta(x, x_1) \leq m(x_1)$, then $x \in d_1$.

It is notable that the pruned rules do not cover all the samples. In addition, the test sample may be beyond the region of the neighborhood of any sample. In this



(a) raw samples (b) spherical neighborhood covering (c) reduced covering

Fig. 10.10. Rule learning with spherical neighborhoods



(a) raw samples (b) quadrate neighborhood covering (c) reduced covering

Fig. 10.11. Rule learning with quadrate neighborhoods

case, no rule matches the test sample. In this case, the test sample is classified to the class of the nearest neighborhood.

As we employ a greedy strategy in searching rules. The rule learning algorithm is very efficient. The time complexities of computing the nearest miss and nearest hit are $n \log n$, respectively. The complexity of the forward rule generation is n. Totally, the time complexity of algorithm NCR is $n \log n$.

We generate two binary classification tasks in 2-D feature spaces. The training samples are shown in Figure 10.10(a) and Figure 10.11(a), respectively. We compute the margin of every sample with Euclidean distance and infinite-norm-based distance, respectively. Then we can build a neighborhood of each sample. Euclidean distance produces spherical neighborhoods and infinite-norm-based distance yields quadrate neighborhoods, as shown in Figure 10.10(b) and Figure 10.11(b), respectively.

Obviously, there are a lot of redundant neighborhoods in the original neighborhood covering. Removing the superfluous covering elements leads to a compact and concise classification model. So we employ Algorithm NCR on the neighborhood covering. The reduced coverings are presented in Figure 10.10(c) and Figure 10.11(c), respectively. As to these simple tasks, two pieces of rules are produced for each task. The classification models are simple and easy to be understood.

10.8 Conclusions and Future Work

Neighborhood structures are very useful for understanding data distribution and building classification models. In this chapter, we introduce the neighborhood rough set model and show some applications in classification learning. This model takes the fundamental assumption in human's cognition that the objects having the same or similar feature values should be grouped into the same class; otherwise, the classification is inconsistent. Here the words "same" and "similar" can be measured with a general distance or similarity function, and the neighborhood of a sample x is a subset of samples which have the same or similar feature values of x. Then we analyze whether all the samples in the neighborhood come from the same class. If all the samples come from the same class, we group x into the positive region; otherwise, group it into classification boundary. By this way, we can divide the samples into two subsets: positive region and boundary region. Using the boundary region, we can train support vector machines; using the positive region, we can evaluate the quality of features for describing classification. By neighborhood covering reduction, we can also extract classification rules from samples. These applications are useful and interesting. Along this way, we can develop new theoretical framework for machine learning. We can also develop new applications of the models.

In the current researches we usually employ Euclidian distance function for numerical data. However, there are a lot of distance functions for more complex data, such as images, time series, graphs and other unstructured data. We can introduce neighborhood rough sets to these domains by employing proper distance functions. We will work along this direction in the future.

References

1. Andersen, T.L., Martinez, T.R.: NP-completeness of minimum rule sets. In: Proceedings of the 10th International Symposium on Computer and Information Sciences, Kusadasi, Aydin, Turkey, pp. 411–418 (1995)
2. Cortes, C., Vapnik, V.N.: Support vector networks. Machine Learning 20, 273–297 (1995)
3. Chen, D.G., Wang, C.Z., Hu, Q.H.: A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets. Information Sciences 177(17), 3500–3518 (2007)
4. Düntsch, I., Gediga, G.: Statistical evaluation of rough set dependency analysis. International Journal of Human Computer Studies 46(5), 589–604 (1997)
5. Fierens, D., Ramon, J., Blockeel, H., Bruynooghe, M.: A comparison of pruning criteria for probability trees. Machine Learning 78, 251–285 (2010)
6. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection - theory and algorithms. In: Proc. 21st Int. Conf. Mach., pp. 43–50. ACM, New York (2004)
7. Hu, X., Cercone, N.: Learning in Relational Databases: A Rough Set Approach. Computational Intelligence 11(2), 323–338 (1995)

8. Hu, Q.H., Pedrycz, W., Yu, D.R., Lang, J.: Selecting discrete and continuous features based on neighborhood decision error minimization. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 40(1), 137–150 (2010)
9. Hu, Q.H., Yu, D.R., Liu, J.F., Wu, C.: Neighborhood rough set based heterogeneous feature subset selection. Information Sciences 178(18), 3577–3594 (2008)
10. Hu, Q.H., Yu, D.R., Xie, Z.X.: Neighborhood classifier. Expert Systems with Applications 34, 866–876 (2008)
11. Hu, Q.H., Zhang, L., Zhang, D., et al.: Measuring relevance between discrete and continuous features based on neighborhood mutual information. Expert Systems with Applications 38, 10737–10750 (2011)
12. He, Q.H., Xie, Z., Hu, Q.H., et al.: Neighborhood based sample and feature selection for SVM classification learning. Neurocomputing 74, 1585–1594 (2011)
13. Du, Y., Hu, Q.H., Zhu, P.F., et al.: Rule learning for classification based on neighborhood covering reduction. Information Sciences 181(24), 5457–5467 (2011)
14. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE Transactions of Knowledge and Data Engineering 16(12), 1457–1471 (2004)
15. Koggalage, R., Halgamuge, S.: Reducing the number of training samples for fast support vector machine classification. Neural Information Processing 2(3), 57–65 (2004)
16. Lee, Y.J., Mangasarian, O.L.: RSVM: Reduced support vector machines. Data Mining Institute Technical Report 00-07 (July 2000); First SIAM International Conference on Data Mining, Chicago (2001)
17. Moshkov, M.J., Skowron, A., Suraj, Z.: On minimal rule sets for almost all binary information systems. Fundamenta Informaticae 80, 247–258 (2008)
18. Palawk, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
19. Pawlak, Z.: Rough Sets–Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordrecht (1991)
20. Ślęzak, D.: Approximate entropy reducts. Fundamenta informaticae 53(3-4), 365–390 (2002)
21. Shin, H., Cho, S.: Invariance of neighborhood relation under input space to feature space mapping. Pattern Recognition Letters 26, 707–718 (2005)
22. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances of Rough Set Theory, pp. 331–362. Kluwer, Dordrecht (1991)
23. Swiniarski, R.W., Skowron, A.: Rough set methods in feature selection and recognition. Pattern Recognition Letters 24, 833–849 (2003)
24. Tsumoto, S.: Automated extraction of hierarchical decision rules from clinical databases using rough set model. Expert Systems with Applications 24(2), 189–197 (2003)
25. Wang, J., Miao, D.Q.: Analysis on attribute reduction strategies of rough sets. Journal of Computer Science and Technology 13(2), 189–192 (1998)
26. Wang, J., Wang, J.: Reduction algorithms based on discernibility matrix: The ordered attributes method. Journal of Computer Science and Technology 16(6), 489–504 (2001)
27. Wang, X.Y., Yang, J., Teng, X.L., et al.: Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters 28(4), 459–471 (2007)
28. Wang, G.Y., Zhao, J., An, J.J., et al.: A comparative study of algebra viewpoint and information viewpoint in attribute reduction. Fundamenta Informaticae 68(3), 289–301 (2005)

29. Ziarko, W.: Variable precision rough sets model. Journal of Computer and System Sciences 46, 39–59 (1993)
30. Zhong, N., Dong, J., Ohsuga, S.: Rule discovery by soft induction techniques. Neurocomputing 36(1-4), 171–204
31. Zhao, M., Han, S.Q., Wang, J.: Tree expressions for information systems. Journal of Computer Science and Technology 22(2), 297–307 (2007)
32. Zhu, W., Wang, F.Y.: On three types of covering-based rough sets. IEEE Transactions on Knowledge and Data Engineering 19(8), 1131–1144 (2007)

Chapter 11
Rough Representations of Ill-Known Sets and Their Manipulations in Low Dimensional Space

Masahiro Inuiguchi

Abstract. Ill-known sets are subsets whose members are not known exactly. They can be represented by a family of subsets that can be true. When each subset is assigned a possible degree, the ill-known set is called a graded ill-known set. In this chapter, we focus on manipulations of graded ill-known sets, a possibility distribution on the power set. Two fuzzy sets on the universe called lower and upper approximations are uniquely defined from a graded ill-known set. On the contrary, a graded ill-known set is not uniquely determined by given lower and upper approximations but the maximal one is. Under a certain condition, we explicitly represent the maximal graded ill-known set having given lower and upper approximations. To utilize graded ill-known sets in decision and information sciences, possibility and necessity measures of graded ill-known sets are described. Simple computation formulae of possibility and necessity measures of graded ill-known sets are shown when lower and upper approximations are given.

Keywords: Ill-known set, fuzzy set, possibility measure, necessity measure, lower approximation, upper approximation, implication function, conjunction function.

11.1 Introduction

Manipulation of uncertain information on single-valued variables has been considerably developed in probability and possibility theories [2, 7, 18, 20, 23]. Along with the development of the theories, the necessity and importance of the treatment of uncertain information on set-valued variables have been recognized [5, 8, 21]. Whereas the knowledge on single-valued variables is said to be disjunctive, the knowledge on set-valued variables is said to be conjunctive. Some frameworks [5, 8, 21] have been already proposed to manipulate conjunctive information. For example, probabilistic information on a set-valued variable can be treated as random sets [5] while

Masahiro Inuiguchi
Graduate School of Engineering Science, 1-3 Machikaneyama-cho, Toyonaka,
Osaka 560-8531, Japan
e-mail: inuiguti@sys.es.osaka-u.ac.jp

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 309–331.
springerlink.com

possibilistic information on a set-valued variable can be represented as ill-known sets [8].

Let us focus on ill-known sets. Ill-known set is a subset of the universal set X whose members are not exactly known. The manipulation of ill-known sets has not yet been discussed exhaustively. The conceivable reasons are (a) we are seldom conscious of the existence of an ill-known set even when we use it, (b) an ill-known set is very complicated because it is a subset of the power set, and thus, (c) manipulation of ill-known sets can be considered as a very formidable task. However, we may treat ill-known sets unconsciously. For example, fuzzy reasoning based on a conjunction function that is well-used in fuzzy logic control [19] can be considered reasoning based on the conjunctive knowledge [10]. In the method, a set of conceivable input values is inferred rather than a single input value. The collections of possibly true objects such as possible facts can be seen as ill-known sets. Someone's favorite food, days when a person stays in Osaka, candidate for research topic, and so on are set-valued variables and the uncertain information about those variables can be represented by ill-known sets. We often encounter ill-known sets in real world. The investigation of ill-known sets would be necessary and important for representing human knowledge and uncertain information.

To define an ill-known set completely, we should specify a family of subsets. Because we have $(2^{|X|} - 1)$ candidates for the member of the family, the specification may become a formidable task when the universe X is large, where $|X|$ shows the cardinality of X. A rough representation is conceivable by specifying two subsets, a subset whose elements are certainly known to be members and a subset whose elements are certainly known not to be members. While the former subset can be seen as a lower approximation of the ill-known set, the complementary subset of the latter subset can be seen as an upper approximation because it is a subset whose elements may be members. In the real world, we often know such lower and upper approximations of ill-known sets but not the complete representation. Most of the time, the lower and upper approximations of ill-known sets would be sufficient and useful. Considering the lower and upper approximations, the rough representations of ill-known sets are similar to rough sets [12, 13]. As the similar set model to the rough representation of ill-known set, shadowed set is studied by Pedrycz [16].

Graded ill-known sets were also considered. A graded ill-known set can be defined by specifying possibility degrees on subsets of X. In other words, a graded ill-known set is represented by a possibility distribution on the power set 2^X. Graded ill-known sets can be approximated by two fuzzy subsets: a fuzzy subset with membership grade showing to what extent the element is a member, and a fuzzy subset with membership grade showing to what extent the element is not a member. Similar to non-graded case, lower and upper approximations of the graded ill-known set are obtained from those two fuzzy subsets. The membership function of lower approximation is assumed to be not greater than that of upper approximation. Therefore, the rough representations of graded ill-known sets are similar to Φ-fuzzy sets [11], intuitionistic fuzzy sets [1] and twofold fuzzy sets [4, 6].

As a pioneer work on ill-known sets and graded ill-known sets, Dubois and Prade [8] gave interesting concepts and presented considerable results. They gave

the fundamental definitions and investigated the recovery of possibility distribution on the power set 2^X from the rough representations and manipulations of possibility and necessity measures of ill-known sets on the universe X. Possibility and necessity measures are useful for the evaluation of vague queries. They assumed Dienes implication and the minimum operation for implication and conjunction functions.

Considering the diversity of people, professions and traditions, Dienes implication and the minimum operation are not always sufficient to represent human evaluations. For example, necessity measure can be seen as a measure of inclusion but we cannot express the normal fuzzy set inclusion [22] by the necessity measure defined by Dienes implication. In approximate reasoning, to characterize many kinds of fuzzy reasoning, many implication and conjunction functions are used. Considering those facts, it would be better to use various implication and conjunction functions in the treatments of ill-known sets.

In this chapter, we focus on the rough representations of graded ill-known sets and the manipulations of possibility and necessity measures of graded ill-known sets using general conjunction and implication functions in the universe X. Although a graded ill-known set can be regarded as a possibility distribution on the power set, the lower and upper approximations enable us to represent it as two fuzzy subsets of the universe. In the real world, because of its complexity, it is not easy to specify a complete possibility distribution of the power set but to specify their lower and upper approximations. We consider graded ill-known sets specified by their upper and lower approximations. There are many possibility distributions on the power set having same lower and upper approximations, but the maximal possibility distribution is unique. Then we take a graded ill-known set corresponding to the maximal possibility distribution. We investigate the manipulations of the maximal graded ill-known sets by their lower and upper approximations. It is shown that possibility and necessity measures are computed by their lower and upper approximations.

In next section, ill-known sets and graded ill-known sets are introduced. Rough representation of ill-known sets and imprecisely specified ill-known sets are investigated. In Section 11.3, possibility and necessity measures of ill-known sets under ill-known information are described. It is shown that those measures with respect to imprecisely specified ill-known sets are calculated in lower-dimensional space. Concluding remarks and several conceivable future topics are described in Section 11.4.

11.2 Ill-Known Sets

Let X be a universe. Let A be a crisp set which is ill-known, that is, there exists at least one element $x \in X$, for which it is not known whether x belongs to A or not. To represent such an ill-known set, collecting possible realizations of A, we obtain the following family:

$$\mathcal{A} = \{A_1, A_2, \ldots, A_n\}, \tag{11.1}$$

where A_i is a crisp set such that $A = A_i$ is possible.

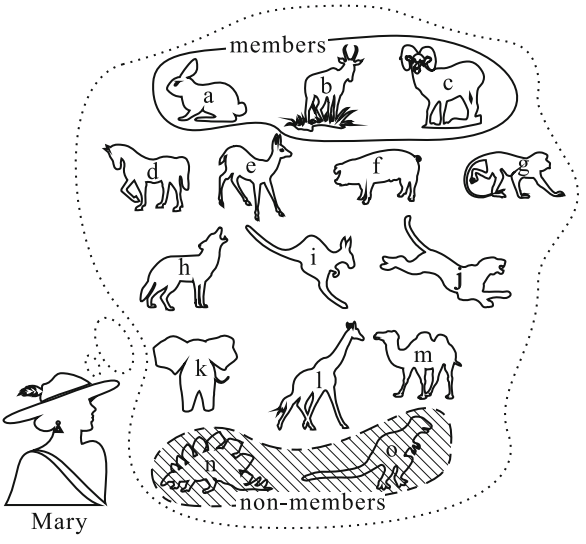


Fig. 11.1. Possible animals in the zoo

Given $\mathcal{A}$, we obtain a set of elements which is a certain member of A, say A^- and a set of elements which is a possible member of A, say A^+ can be defined as

$$A^- = \bigcap \mathcal{A} = \bigcap_{i=1,\ldots,n} A_i, \quad A^+ = \bigcup \mathcal{A} = \bigcup_{i=1,\ldots,n} A_i. \tag{11.2}$$

Corresponding to rough sets [12, 13], we call A^- and A^+ "the lower approximation" of A and "the upper approximation" of A, respectively.

In the real world, we may know certain members and certain non-members only. In other words, we know the lower approximation A^- as a set of certain members and the upper approximation A^+ as a complementary set of certain non-members. Given A^- and A^+ (or equivalently, the complement of A^+), we obtain a family $\mathcal{A}$ of possible realizations of A as

$$\mathcal{A} = \{A_i \mid A^- \subseteq A_i \subseteq A^+\}. \tag{11.3}$$

We note that A^- and A^+ are recovered by applying (11.2) to the family $\mathcal{A}$ induced from A^- and A^+ by (11.3). On the other hand, $\mathcal{A}$ cannot be always recovered by applying (11.3) to A^- and A^+ defined by (11.2).

Example 11.1. Mary and her son are going to a small zoo where she visited 10 years ago. Let us consider the animals they can see in the zoo. Mary remembers that there were rabbits (a), goats (b) and sheep (c).

Let $X = \{\mathrm{a}, \mathrm{b}, \ldots, \mathrm{o}\}$ be a set including all possible animals (see Figure 11.1). Let A be the set of animals they can see. From her memory, Mary knows that A includes

$\{a,b,c\}$. Moreover, Mary knows that dinosaurs are extinct. Namely, A include neither $\{n\}$ nor $\{o\}$. However, Mary does not know about whether the other animals they can see at the zoo or not. Then A is an ill-known set. From Mary's memory and the general knowledge, we know that A is a set including neither $\{n\}$ nor $\{o\}$ but $\{a,b,c\}$. In other words, we have $A^- = \{a,b,c\}$ as the lower approximation and $A^+ = X - \{n,o\}$ as the upper approximation of A. Then we obtain the family of possible realizations as

$$\mathcal{A} = \{A_i \mid A_i \supseteq \{a,b,c\},\ A_i \cap \{n,o\} = \emptyset\}. \tag{11.4}$$

Obviously, applying (11.2) to $\mathcal{A}$ of (11.4), we regain $A^- = \{a,b,c\}$ and $A^+ = X - \{n,o\}$.

If all A_i's of (11.1) are not regarded as equally possible, we may assign a possibility degree $\pi_{\mathcal{A}}(A_i)$ to each A_i so that

$$\max_{i=1,\ldots,n} \pi_{\mathcal{A}}(A_i) = 1. \tag{11.5}$$

Thus, a possibility distribution $\pi_{\mathcal{A}} : 2^X \to [0,1]$ is introduced. The ill-known set having such a possibility distribution is called "a graded ill-known set".

In this case, the lower approximation A^- and the upper approximation A^+ are defined as fuzzy sets with the following membership functions:

$$\mu_{A^-}(x) = \inf_{x \notin A_i} I_1(\pi_{\mathcal{A}}(A_i),\ 0), \tag{11.6}$$

$$\mu_{A^+}(x) = \sup_{x \in A_i} T_1(\pi_{\mathcal{A}}(A_i),\ 1), \tag{11.7}$$

where I_1 and T_1 are implication and conjunction functions, respectively. An implication function $I : [0,1] \times [0,1] \to [0,1]$ satisfies

(I1) $I(0,0) = I(0,1) = I(1,1) = 1$ and $I(1,0) = 0$, (boundary condition)
(I2) $I(a,b) \le I(c,d),\ 0 \le c \le a \le 1,\ 0 \le b \le d \le 1.$ (monotonicity)

A conjunction function $T : [0,1] \times [0,1] \to [0,1]$ satisfies

(T1) $T(0,0) = T(0,1) = T(1,0) = 0$ and $T(1,1) = 1$, (boundary condition)
(T2) $T(a,b) \le T(c,d),\ 0 \le a \le c \le 1,\ 0 \le b \le d \le 1.$ (monotonicity)

In (11.6) and (11.7), we use functions $I_1(\cdot,0)$ and $T_1(\cdot,1)$ defined by implication and conjunction functions, respectively. However, they can be replaced with a non-increasing function $n : [0,1] \to [0,1]$ and a non-decreasing function $\varphi : [0,1] \to [0,1]$ such that $n(0) = \varphi(1) = 1$ and $n(1) = \varphi(0) = 0$. Indeed, Dubois and Prade [8] defined those by $I_1(s,0) = 1 - s$ and $T_1(s,1) = s$, $\forall s \in [0,1]$. In this chapter, in order to know the relations of those functions with implication and conjunction functions used for other purposes, we use I_1 and T_1 to define A^- and A^+.

Example 11.2. (a continuation of Example 1) Mary asked some people who visited the zoo about animals there. However, as shown in Figure 11.2, their answers are

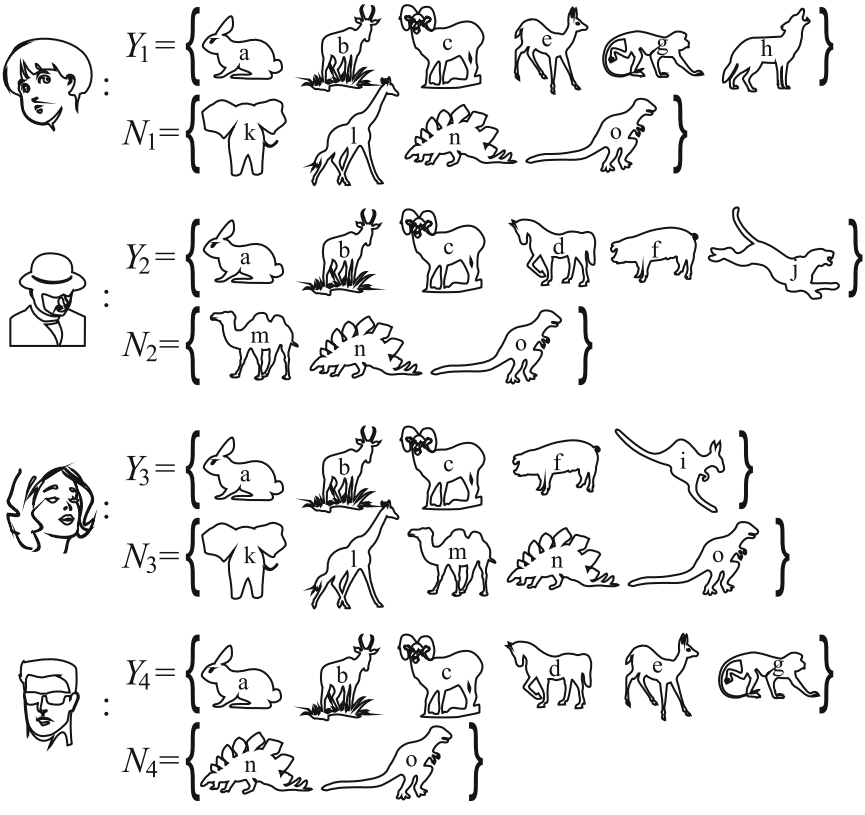


Fig. 11.2. Four opinions about animals in the zoo and Mary's assignment of possibility degrees

different from one another. They answered certain members Y_i and certain nonmembers N_i as $Y_1 = \{\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{e},\mathrm{g},\mathrm{h}\}$, $N_1 = \{\mathrm{k},\mathrm{l},\mathrm{n},\mathrm{o}\}$, $Y_2 = \{\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d},\mathrm{f},\mathrm{j}\}$, $N_2 = \{\mathrm{m},\mathrm{n},\mathrm{o}\}$, $Y_3 = \{\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{f},\mathrm{i}\}$, $N_3 = \{\mathrm{k},\mathrm{l},\mathrm{m},\mathrm{n},\mathrm{n}\}$, $Y_4 = \{\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d},\mathrm{e},\mathrm{g}\}$, and $N_4 = \{\mathrm{n},\mathrm{o}\}$. Those four kinds of information are different, but they are not totally conflicting because we have $Y_i \cap N_j = \emptyset$ for any $i, j \in \{1,2,3,4\}$. Based on those pieces of information, she may assign possibility degrees of

$$\pi_{\mathcal{A}}(A) = \frac{|\{i \in \{1,2,3,4\} \mid A \supseteq Y_i,\ A \cap N_i = \emptyset\}|}{4}. \tag{11.8}$$

Possibility degree $\pi_{\mathcal{A}}(A)$ of (11.8) shows what percentage of people are consistent with A.

Let $T_1(a,b) = \min(a,b)$ and $I_1(a,b) = \max(1-a,b)$. Then we have $T_1(a,1) = a$ and $I_1(a,0) = 1-a$. Applying (11.6) and (11.7), we obtain the following result (see Figure 11.3):

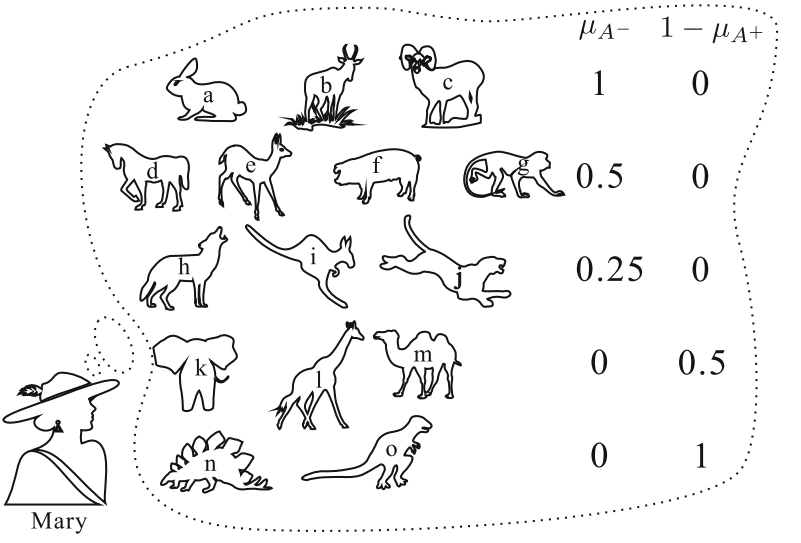


Fig. 11.3. lower and upper approximations of the graded ill-known set in Example 2

$$\begin{aligned}
&\mu_{A^-}(\mathrm{a}) = 1,\ \mu_{A^+}(\mathrm{a}) = 1,\ \mu_{A^-}(\mathrm{b}) = 1,\ \mu_{A^+}(\mathrm{b}) = 1,\\
&\mu_{A^-}(\mathrm{c}) = 1,\ \mu_{A^+}(\mathrm{c}) = 1,\ \mu_{A^-}(\mathrm{d}) = 0.5,\ \mu_{A^+}(\mathrm{d}) = 1,\\
&\mu_{A^-}(\mathrm{e}) = 0.5,\ \mu_{A^+}(\mathrm{e}) = 1,\ \mu_{A^-}(\mathrm{f}) = 0.5,\ \mu_{A^+}(\mathrm{f}) = 1,\\
&\mu_{A^-}(\mathrm{g}) = 0.5,\ \mu_{A^+}(\mathrm{g}) = 1,\ \mu_{A^-}(\mathrm{h}) = 0.25,\ \mu_{A^+}(\mathrm{h}) = 1,\\
&\mu_{A^-}(\mathrm{i}) = 0.25,\ \mu_{A^+}(\mathrm{i}) = 1,\ \mu_{A^-}(\mathrm{j}) = 0.25,\ \mu_{A^+}(\mathrm{j}) = 1,\\
&\mu_{A^-}(\mathrm{k}) = 0,\ \mu_{A^+}(\mathrm{k}) = 0.5,\ \mu_{A^-}(\mathrm{l}) = 0,\ \mu_{A^+}(\mathrm{l}) = 0.5,\\
&\mu_{A^-}(\mathrm{m}) = 0,\ \mu_{A^+}(\mathrm{m}) = 0.5,\ \mu_{A^-}(\mathrm{n}) = 0,\ \mu_{A^+}(\mathrm{n}) = 0,\\
&\mu_{A^-}(\mathrm{o}) = 0,\ \mu_{A^+}(\mathrm{o}) = 0. \qquad (11.9)
\end{aligned}$$

In Example 2, we assume $Y_i \cap N_j = \emptyset$ for any $i, j \in \{1,2,3,4\}$. If this assumption does not hold, the information pieces from the four people are strongly conflicting and we cannot represent all information pieces as a normal possibility distribution $\pi_{\mathcal{A}}$. In such cases, we should apply a certain conflict resolution technique. However, as in Example 2, if information pieces were about a real fact, conflicts would not occur frequently.

We obtain the following proposition about lower and upper approximations.

Proposition 11.1. *We have*

$$\forall x \in X,\ \mu_{A^-}(x) > 0 \text{ implies } \mu_{A^+}(x) = 1. \qquad (11.10)$$

In other words, we have

$$\mathrm{Supp}(A^-) \subseteq \mathrm{Core}(A^+), \tag{11.11}$$

where we define

$$\mathrm{Supp}(A^-) = \{x \in X \mid \mu_{A^-}(x) > 0\}, \quad \mathrm{Core}(A^+) = \{x \in X \mid \mu_{A^+}(x) = 1\}. \tag{11.12}$$

Proof. $\mu_{A^-}(x) > 0$, that is, $\inf_{x \notin A_i} I_1(\pi_{\mathcal{A}}(A_i), 0) > 0$ implies

$$\forall i \in \{1, \ldots, n\} \text{ such that } x \notin A_i,\ \pi_{\mathcal{A}}(A_i) \neq 1.$$

From (11.5), this means

$$\exists i \in \{1, \ldots, n\} \text{ such that } x \in A_i,\ \pi_{\mathcal{A}}(A_i) = 1.$$

Hence, we have

$$\mu_{A^+}(x) = \sup_{x \in A_i} T_1(\pi_{\mathcal{A}}(A_i), 1) \geq T_1(1,1) = 1. \qquad (Q.E.D.)$$

From Proposition 11.1, pair (A^-, A^+) can be seen as a twofold fuzzy set [4, 6] that imposes that $\mu_{A^-}(x) > 0$ implies $\mu_{A^+}(x) = 1$.

Now let us discuss the inverse problem. Namely, when μ_{A^-} and μ_{A^+} are given, let us consider the associated ill-known set, that is, the associated possibility distribution $\pi_{\mathcal{A}} : 2^X \to [0,1]$. Then, the system of equations is given as

$$\mu_{A^-}(x) = \inf_{x \notin A_i} I_1(\pi_{\mathcal{A}}(A_i),\ 0), \quad \mu_{A^+}(x) = \sup_{x \in A_i} T_1(\pi_{\mathcal{A}}(A_i),\ 1). \tag{11.13}$$

Considering that approximations A^- and A^+ would be obtained more often than the possibility distribution $\pi_{\mathcal{A}}$, this inverse problem is practically important.

The existence of a solution which satisfies (11.13) for any given μ_{A^-} and μ_{A^+} requires the following conditions on I_1 and T_1:

$$\{I_1(s,0) \mid s \in [0,1]\} = [0,1], \quad \{T_1(s,1) \mid s \in [0,1]\} = [0,1]. \tag{11.14}$$

Taking into consideration the definitions of implication and conjunction functions, (11.14) implies that the functions $I_1(\cdot,0) : [0,1] \to [0,1]$ and $T_1(\cdot,1) : [0,1] \to [0,1]$ are decreasing and increasing surjections, respectively. This also means that $I_1(\cdot,0)$ and $T_1(\cdot,1)$ are continuous.

We have the following proposition on the greatest possibility distribution $\pi_{\mathcal{A}}$ satisfying (11.13), where we say that $\pi_{\mathcal{A}_1}$ is not smaller than $\pi_{\mathcal{A}_2}$ if and only if $\pi_{\mathcal{A}_1}(A_i) \geq \pi_{\mathcal{A}_2}(A_i), \forall A_i \in 2^X$.

Proposition 11.2. *When (11.14) is fulfilled, the maximal possibility distribution satisfying (11.13) is obtained uniquely as*

$$\pi^*_{\mathcal{A}}(A_i) = \begin{cases} \inf\limits_x I_1^*(\mu_{A^-}(x),0), & \text{if } A_i = \emptyset, \\ \inf\limits_x T_1^*(\mu_{A^+}(x),1), & \text{if } A_i = X, \\ \min\left(\inf\limits_{x\notin A_i} I_1^*(\mu_{A^-}(x),0), \inf\limits_{x\in A_i} T_1^*(\mu_{A^+}(x),1) \right), & \text{otherwise,} \end{cases} \tag{11.15}$$

where

$$I_1^*(a,0) = \sup\{s\in[0,1] \mid I_1(s,0) \geq a\}, \tag{11.16}$$
$$T_1^*(a,1) = \sup\{s\in[0,1] \mid T_1(s,1) \leq a\}. \tag{11.17}$$

Proof. From the assumption of the proposition, (11.14) is satisfied with I_1 and T_1. For a solution $\pi_{\mathcal{A}}$ satisfying (11.13), we have

$$\pi_{\mathcal{A}}(A_i) \leq \sup\{s\in[0,1] \mid I_1(s,0) \geq \mu_{A^-}(x)\},\ \forall x\notin A_i, \text{for } A_i \neq X,$$
$$\pi_{\mathcal{A}}(A_i) \leq \sup\{s\in[0,1] \mid T_1(s,1) \leq \mu_{A^+}(x)\},\ \forall x\in A_i, \text{for } A_i \neq \emptyset.$$

Applying (11.16) and (11.17), we obtain

$$\pi_{\mathcal{A}}(A_i) \leq \inf_{x\notin A_i} I_1^*(\mu_{A^-}(x),0), \quad \text{for } A_i \neq X,$$
$$\pi_{\mathcal{A}}(A_i) \leq \inf_{x\in A_i} T_1^*(\mu_{A^+}(x),1), \quad \text{for } A_i \neq \emptyset.$$

Combining those, for a solution $\pi_{\mathcal{A}}$ satisfying (11.13), we have

$$\pi_{\mathcal{A}}(A_i) \leq \pi^*_{\mathcal{A}}(A_i).$$

Therefore, it is sufficient to show $\pi^*_{\mathcal{A}}$ of (11.15) satisfies

$$\mu_{A^-}(x) = \inf_{x\notin A_i} I_1(\pi^*_{\mathcal{A}}(A_i),0), \quad \mu_{A^+}(x) = \sup_{x\in A_i} T_1(\pi^*_{\mathcal{A}}(A_i),1). \tag{11.18}$$

First, let us prove the latter equality of (11.18). For $x\notin \mathrm{Supp}(A^-)$, we have

$$\pi^*_{\mathcal{A}}(\mathrm{Supp}(A^-)\cup\{x\})$$
$$= \min(I_1^*(0,0), \min(T_1^*(1,1), T_1^*(\mu_{A^+}(x),1))) = T_1^*(\mu_{A^+}(x),1)$$

because we have $\mu_{A^+}(y) = 1$ for $y\in\mathrm{Supp}(A^-)$ by Proposition 11.1. Then we obtain

$$\sup_{x\in A_i} T_1(\pi^*_{\mathcal{A}}(A_i),1) \geq T_1(\pi^*_{\mathcal{A}}(\mathrm{Supp}(A^-)\cup\{x\}),1)$$
$$= T_1(T_1^*(\mu_{A^+}(x),1),1)$$
$$= \sup\{T_1(s,1) \mid T_1(s,1) \leq \mu_{A^+}(x),\ s\in[0,1]\} = \mu_{A^+}(x).$$

On the other hand, since, for any A_i such that $x\in A_i$, $\pi^*_{\mathcal{A}}(A_i) \leq T_1^*(\mu_{A^+}(x),1)$ is valid, we have

$$T_1(\pi^*_{\mathcal{A}}(A_i),1) \leq T_1(T_1^*(\mu_{A^+}(x),1),1) \leq \mu_{A^+}(x), \quad \forall A_i \text{ such that } x\in A_i.$$

Hence, the latter part of (11.18) is proved when $x \notin \mathrm{Supp}(A^-)$.

For $x \in \mathrm{Supp}(A^-) \subseteq \mathrm{Core}(A^+)$ (the inclusion comes from Proposition 11.1), since $\pi^*_{\mathcal{A}}(\mathrm{Supp}(A^-)) = 1$, we have

$$\sup_{x \in A_i} T_1(\pi^*_{\mathcal{A}}(A_i), 1) \geq T_1(\pi^*_{\mathcal{A}}(\mathrm{Supp}(A^-)), 1) = 1.$$

Hence, $\sup_{x \in A_i} T_1(\pi^*_{\mathcal{A}}(A_i), 1) = 1 = \mu_{A^+}(x)$.

Now let us prove the former part of (11.18). For $x \in \mathrm{Core}(A^+)$, from Proposition 11.1, we obtain $\pi^*_{\mathcal{A}}(\mathrm{Core}(A^+) - \{x\}) = I_1^*(\mu_{A^-}(x), 0)$. Then,

$$\begin{aligned}\inf_{x \notin A_i} I_1(\pi^*_{\mathcal{A}}(A_i), 0) &\leq I_1(\pi^*_{\mathcal{A}}(\mathrm{Core}(A^+) - \{x\}), 0) \\ &= I_1(I_1^*(\mu_{A^-}(x), 0), 0) \\ &= \inf\{I_1(s, 0) \mid I_1(s, 0) \geq \mu_{A^-}(x)\} = \mu_{A^-}(x).\end{aligned}$$

On the other hand, since, for all A_i such that $x \notin A_i$, $\pi^*_{\mathcal{A}}(A_i) \leq I_1^*(\mu_{A^-}(x), 0)$ is valid, we have

$$I_1(\pi^*_{\mathcal{A}}(A_i), 0) \geq I_1(I_1^*(\mu_{A^-}(x), 0), 0) \geq \mu_{A^-}(x), \quad \forall A_i \text{ such that } x \in A_i.$$

Hence, the former part of (11.18) is proved when $x \in \mathrm{Core}(A^+)$.

For $x \notin \mathrm{Core}(A^+)$, since $\pi^*_{\mathcal{A}}(\mathrm{Core}(A^+)) = 1$, we have

$$\inf_{x \notin A_i} I_1(\pi^*_{\mathcal{A}}(A_i), 0) \geq I_1(\pi^*_{\mathcal{A}}(\mathrm{Core}(A^+)), 0) = 0.$$

Hence, $\inf_{x \notin A_i} I_1(\pi^*_{\mathcal{A}}(A_i), 0) = 0 = \mu_{A^-}(x)$. (Q.E.D.)

In this chapter, we introduce the following assumption which is a little bit stronger than (11.14):

(A1) $I_1(\cdot, 0) : [0,1] \to [0,1]$ and $T_1(\cdot, 1) : [0,1] \to [0,1]$ are decreasing and increasing bijection, respectively.

Under assumption (A1), we have $I_1^*(\cdot, 0) = I_1^{-1}(\cdot, 0)$ and $T_1^*(\cdot, 0) = T_1^{-1}(\cdot, 0)$, where $I_1^{-1}(\cdot, 0)$ and $T_1^{-1}(\cdot, 0)$ are inverse functions of $I_1(\cdot, 0)$ and $T_1(\cdot, 0)$, respectively. Assuming $\inf \emptyset = +\infty$, (11.15) is simplified as

$$\pi^*_{\mathcal{A}}(A_i) = \min\left(\inf_{x \notin A_i} I_1^{-1}(\mu_{A^-}(x), 0), \inf_{x \in A_i} T_1^{-1}(\mu_{A^+}(x), 1)\right). \tag{11.19}$$

We usually have many solutions which satisfy (11.14). This means that we cannot always recover the original ill-known set. As Dubois and Prade [8] mentioned, an ill-known set represented as a possibility distribution on the power set 2^X is too sophisticated to be dealt with. It is rare that an ill-known set is given by a complete possibility distribution form. In the real world, we may have cases where only lower and upper approximations A^- and A^+ are obtained rather than the complete possibility distribution $\pi_{\mathcal{A}}$. In this chapter, we focus on the manipulation of a graded

ill-known set imprecisely specified by its approximations because it would be more encountered in the real world than that specified by its complete possibility distribution. We call the imprecise specification of an ill-known set by lower and upper approximations satisfying (11.10) "a rough representation of ill-known set".

The determination of a possibility distribution on 2^X corresponding to the lower and upper approximations is called the "embedding". For a rough representation of ill-known set, the given approximations are recovered through the embedding and approximation processes as shown in Proposition 11.2.

The complement of an ill-known set expressed by a possibility distribution $\pi_{\mathcal{A}}$ is expressed by a possibility distribution $\bar{\pi}_{\bar{\mathcal{A}}}$ defined by

$$\bar{\pi}_{\bar{\mathcal{A}}}(\bar{A}_i) = \pi_{\mathcal{A}}(A_i),\ i = 1,\ldots,n, \tag{11.20}$$

where $\bar{A}_i$ is the complement of A_i.

Under assumption (A1), the lower and upper approximations of the complementary set are given by the following lower and upper approximations of an ill-known set:

$$\begin{aligned}
\mu_{\bar{A}^+}(x) &= \sup_{x\in\bar{A}_i} T_1(\bar{\pi}_{\bar{\mathcal{A}}}(\bar{A}_i),1) = \sup_{x\notin A_i} T_1(\pi_{\mathcal{A}}(A_i),1)\\
&= \sup_{x\notin A_i} T_1(I_1^{-1}(I_1(\pi_{\mathcal{A}}(A_i),0),0),1)\\
&= T_1\left(I_1^{-1}\left(\inf_{x\notin A_i} I_1(\pi_{\mathcal{A}}(A_i),0),0\right),1\right) = T_1(I_1^{-1}(\mu_{A^-}(x),0),1), \qquad (11.21)\\
\mu_{\bar{A}^-}(x) &= \inf_{x\notin\bar{A}_i} I_1(\bar{\pi}_{\bar{\mathcal{A}}}(\bar{A}_i),0) = \inf_{x\in A_i} I_1(\pi_{\mathcal{A}}(A_i),0)\\
&= \inf_{x\in A_i} I_1(T_1^{-1}(T_1(\pi_{\mathcal{A}}(A_i),1),1),0)\\
&= I_1\left(T_1^{-1}\left(\sup_{x\in A_i} T_1(\pi_{\mathcal{A}}(A_i),1),1\right),0\right) = I_1(T_1^{-1}(\mu_{A^+}(x),1),0), \qquad (11.22)
\end{aligned}$$

The relations among $\pi_{\mathcal{A}}$, $\bar{\pi}_{\bar{\mathcal{A}}}$ and μ_{A^-}, μ_{A^+}, $\mu_{\bar{A}^-}$, $\mu_{\bar{A}^+}$ under assumption (A1) are depicted in Figure 11.4.

Example 11.3. (a continuation of Example 2) Let $T_1(a,1) = a$ and $I_1(a,0) = 1-a$ as in Example 2. Then $T_1^{-1}(a,1) = a$ and $I_1^{-1}(a,0) = 1-a$. Let $P_1 = \{\mathrm{a},\mathrm{b},\mathrm{c}\}$, $P_2 = \{\mathrm{d},\mathrm{e},\mathrm{f},\mathrm{g}\}$, $P_3 = \{\mathrm{h},\mathrm{i},\mathrm{j}\}$, $P_4 = \{\mathrm{k},\mathrm{l},\mathrm{m}\}$ and $P_5 = \{\mathrm{n},\mathrm{o}\}$. From μ_{A^-} and μ_{A^+} given in (11.9), we obtain

$$\pi^*_{\mathcal{A}}(A) = \begin{cases} 0, & \text{if } A \not\supseteq P_1 \text{ or } A\cap P_5 \neq \emptyset,\\ 0.5, & \text{if } A \supseteq P_1, A\cap P_5 = \emptyset \text{ and } (A \not\supseteq P_2 \text{ or } A\cap P_4 \neq \emptyset),\\ 0.75, & \text{if } A \supseteq P_1\cup P_2, A \not\supseteq P_3 \text{ and } A\cap(P_4\cup P_5) = \emptyset,\\ 1, & \text{if } A \supseteq P_1\cup P_2\cup P_3 \text{ and } A\cap(P_4\cup P_5) = \emptyset. \end{cases} \tag{11.23}$$

In this case, $\pi^*_{\mathcal{A}}$ is different from $\pi_{\mathcal{A}}$ defined by (11.8). Indeed, for $A = \{\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{m}\}$, we obtain $\pi^*_{\mathcal{A}}(A) = 0.5$ while $\pi_{\mathcal{A}}(A) = 0$.

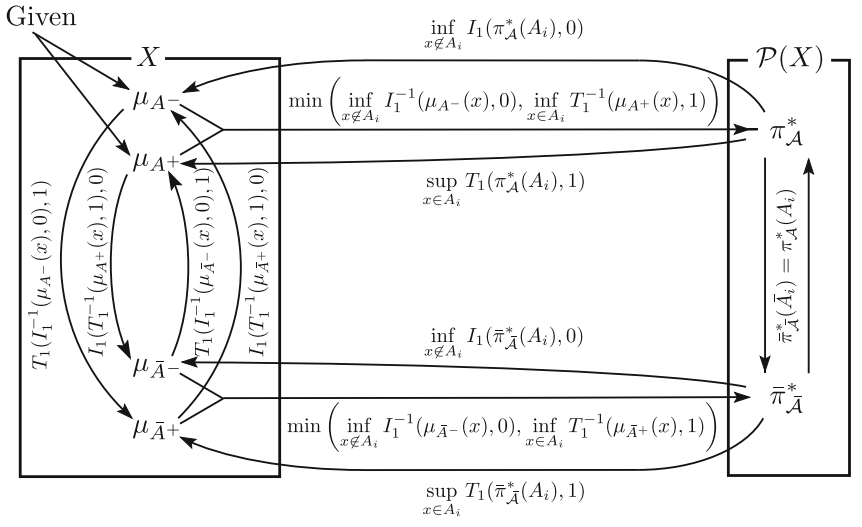


Fig. 11.4. Relations among $\pi_{\mathcal{A}}$, $\bar{\pi}_{\bar{\mathcal{A}}}$ and $\mu_{A^-}, \mu_{A^+}, \mu_{\bar{A}^-}, \mu_{\bar{A}^+}$ under assumption (A1)

The complement of the ill-known set imprecisely specified by μ_{A^-} and μ_{A^+} of (11.9) is an ill-known set imprecisely specified by the following $\mu_{\bar{A}^-}$ and $\mu_{\bar{A}^+}$:

$$\begin{aligned}
&\mu_{\bar{A}^-}(\mathrm{a})=0,\ \mu_{\bar{A}^+}(\mathrm{a})=0,\ \mu_{\bar{A}^-}(\mathrm{b})=0,\ \mu_{\bar{A}^+}(\mathrm{b})=0,\\
&\mu_{\bar{A}^-}(\mathrm{c})=0,\ \mu_{\bar{A}^+}(\mathrm{c})=0,\ \mu_{\bar{A}^-}(\mathrm{d})=0,\ \mu_{\bar{A}^+}(\mathrm{d})=0.5,\\
&\mu_{\bar{A}^-}(\mathrm{e})=0,\ \mu_{\bar{A}^+}(\mathrm{e})=0.5,\ \mu_{\bar{A}^-}(\mathrm{f})=0,\ \mu_{\bar{A}^+}(\mathrm{f})=0.5,\\
&\mu_{\bar{A}^-}(\mathrm{g})=0,\ \mu_{\bar{A}^+}(\mathrm{g})=0.5,\ \mu_{\bar{A}^-}(\mathrm{h})=0,\ \mu_{\bar{A}^+}(\mathrm{h})=0.75,\\
&\mu_{\bar{A}^-}(\mathrm{i})=0,\ \mu_{\bar{A}^+}(\mathrm{i})=0.75,\ \mu_{\bar{A}^-}(\mathrm{j})=0,\ \mu_{\bar{A}^+}(\mathrm{j})=0.75,\\
&\mu_{\bar{A}^-}(\mathrm{k})=0.5,\ \mu_{\bar{A}^+}(\mathrm{k})=1,\ \mu_{\bar{A}^-}(\mathrm{l})=0.5,\ \mu_{\bar{A}^+}(\mathrm{l})=1,\\
&\mu_{\bar{A}^-}(\mathrm{m})=0.5,\ \mu_{\bar{A}^+}(\mathrm{m})=1,\ \mu_{\bar{A}^-}(\mathrm{n})=1,\ \mu_{\bar{A}^+}(\mathrm{n})=1,\\
&\mu_{\bar{A}^-}(\mathrm{o})=1,\ \mu_{\bar{A}^+}(\mathrm{o})=1.
\end{aligned} \tag{11.24}$$

11.3 Possibility and Necessity Measures under Ill-Known Sets

In the presence of a partial information given by a rough representation of ill-known set $\mathcal{A}$, let us consider the evaluation of a vague query expressed by a rough representation of ill-known set $\mathcal{B}$. Since the information has been given as a rough representation of ill-known set $\mathcal{A}$, we can obtain a possibility distribution $\pi_{\mathcal{A}}$ by embedding formula (11.15). Similarly, the vague query given by $\mathcal{B}$ can be expressed as a fuzzy set characterized by $\pi_{\mathcal{B}}$. Then, as is investigated by Prade and Testemale [17], the evaluation can be done by the possibility and necessity measures of $\mathcal{B}$ under $\pi_{\mathcal{A}}$. The

possibility distribution $\pi_{\mathcal{A}}$ is defined on the power set 2^X, and thus, it may require the calculations on the power set 2^X. We investigate the calculations of possibility and necessity measures using the lower and upper approximations A^-, A^+, B^- and B^+.

Possibility and necessity measures of an ill-known set $\mathcal{B}$ under an ill-known set $\mathcal{A}$ on the power set can be defined by

$$\Pi(\mathcal{B}|\mathcal{A}) = \sup_{C\in 2^X} T_2(\pi_{\mathcal{A}}(C), \pi_{\mathcal{B}}(C)), \tag{11.25}$$

$$N(\mathcal{B}|\mathcal{A}) = \inf_{C\in 2^X} I_2(\pi_{\mathcal{A}}(C), \pi_{\mathcal{B}}(C)), \tag{11.26}$$

where T_2 and I_2 are conjunction and implication functions, respectively. Those measures evaluate possibility and necessity degrees to what extent the set expressed as $\mathcal{A}$ possibly equals to or possibly matches to the set expressed as $\mathcal{B}$.

When $\mathcal{A}$ and $\mathcal{B}$ are given roughly by their approximations, we obtain the following proposition on calculations of necessity measures.

Proposition 11.3. *Under the assumption (A1), when I_2 is upper semi-continuous, we have*

$$N(\mathcal{B}|\mathcal{A}) = \min\left(\inf_{x\in X} I_3(\mu_{A^+}(x), \mu_{B^+}(x)), \inf_{x\in X} I_4(\mu_{B^-}(x), \mu_{A^-}(x))\right), \tag{11.27}$$

where implication functions I_3 and I_4 are defined by

$$I_3(a,b) = I_2\left(T_1^{-1}(a,1), T_1^{-1}(b,1)\right), \tag{11.28}$$

$$I_4(a,b) = I_2\left(I_1^{-1}(b,0), I_1^{-1}(a,0)\right). \tag{11.29}$$

Proof. Utilizing the upper semi-continuity of I_2 and assumption (A1), this proposition is proved as

$$\begin{aligned}
N(\mathcal{B}|\mathcal{A}) &= \inf_{C\in 2^X} I_2(\pi^*_{\mathcal{A}}(C), \pi^*_{\mathcal{B}}(C)) \\
&= \inf_{C\in 2^X} I_2\left(\pi^*_{\mathcal{A}}(C), \min\left(\inf_{x\in C} T_1^{-1}(\mu_{B^+}(x),1), \inf_{x\notin C} I_1^{-1}(\mu_{B^-}(x),0)\right)\right) \\
&= \min\left(\inf_{C\in 2^X} I_2\left(\pi^*_{\mathcal{A}}(C), \inf_{x\in C} T_1^{-1}(\mu_{B^+}(x),1)\right),\right. \\
&\qquad \left.\inf_{C\in 2^X} I_2\left(\pi^*_{\mathcal{A}}(C), \inf_{x\notin C} I_1^{-1}(\mu_{B^-}(x),0)\right)\right) \\
&= \min\left(\inf_{C\in 2^X}\inf_{x\in C} I_2\left(\pi^*_{\mathcal{A}}(C), T_1^{-1}(\mu_{B^+}(x),1)\right),\right. \\
&\qquad \left.\inf_{C\in 2^X}\inf_{x\notin C} I_2\left(\pi^*_{\mathcal{A}}(C), I_1^{-1}(\mu_{B^-}(x),0)\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&= \min\left(\inf_{x\in C} I_2\left(\sup_{C\in 2^X} \pi^*_{\mathcal{A}}(C), T_1^{-1}(\mu_{B^+}(x),1)\right),\right.\\
&\qquad\left.\inf_{x\notin C} I_2\left(\sup_{C\in 2^X} \pi^*_{\mathcal{A}}(C), I_1^{-1}(\mu_{B^-}(x),0)\right)\right)\\
&= \min\left(\inf_{x\in C} I_2\left(T_1^{-1}\left(\sup_{C\in 2^X} T_1(\pi^*_{\mathcal{A}}(C),1),1\right), T_1^{-1}(\mu_{B^+}(x),1)\right),\right.\\
&\qquad\left.\inf_{x\notin C} I_2\left(I_1^{-1}\left(\inf_{C\in 2^X} I_1(\pi^*_{\mathcal{A}}(C),0),0\right), I_1^{-1}(\mu_{B^-}(x),0)\right)\right)\\
&= \min\left(\inf_{x\in X} I_3(\mu_{A^+}(x),\mu_{B^+}(x)), \inf_{x\in X} I_4(\mu_{B^-}(x),\mu_{A^-}(x))\right).
\end{aligned}
$$

(Q.E.D.)

Before describing a similar result to Proposition 11.3 about possibility measures, we show the following lemma.

Lemma 11.1. *When $\mathcal{A}$ is characterized by possibility distribution $\pi^*_{\mathcal{A}}$ of (11.15), the level cut $[\mathcal{A}]_h = \{C \in 2^X \mid \pi^*_{\mathcal{A}}(C) \geq h\}$ is represented as*

$$
[\mathcal{A}]_h = \left\{C \;\middle|\; (A^-)_{I_1(h,0)} \subseteq C \subseteq [A^+]_{T_1(h,1)}\right\}, \tag{11.30}
$$

where

$$
(A^-)_{I_1(h,0)} = \{x \in X \mid \mu_{A^-}(x) > I_1(h,0)\}, \tag{11.31}
$$

$$
[A^+]_{T_1(h,1)} = \{x \in X \mid \mu_{A^+}(x) \geq T_1(h,1)\}. \tag{11.32}
$$

Proof. Let C be a set such that $(A^-)_{I_1(h,0)} \subseteq C \subseteq [A^+]_{T_1(h,1)}$. Then, we have

$$
\begin{aligned}
&\mu_{A^-}(x) \leq I_1(h,0) \text{ for any } x \notin C,\\
&\mu_{A^+}(x) \geq T_1(h,1) \text{ for any } x \in C.
\end{aligned}
$$

From (I2) and (T2), we obtain

$$
\begin{aligned}
&I_1^*(\mu_{A^-}(x),0) \geq I_1^*(I_1(h,0),0) \geq h, \text{ for any } x \notin C,\\
&T_1^*(\mu_{A^+}(x),1) \geq T_1^*(T_1(h,1),1) \geq h.
\end{aligned}
$$

As a result, we have

$$
\inf_{x\notin C} I_1^*(\mu_{A^-}(x),0) \geq h \text{ and } \inf_{x\in C} T_1^*(\mu_{A^+}(x),1) \geq h.
$$

From (11.15), we obtain $\pi_{\mathcal{A}}(C)^* \geq h$.

Now, utilizing the following equivalences, we prove $\pi^*_{\mathcal{A}}(C) < h$ for any C not satisfying $(A^-)_{I_1(h,0)} \subseteq C \subseteq [A^+]_{T_1(h,1)}$. Because $I_1(\cdot,0)$ is surjective, we obtain

$$I_1^*(a,0) < I_1^*(b,0) \text{ if and only if } a > b. \tag{11.33}$$

Similarly, because $T_1(\cdot,1)$ is surjective, we have

$$T_1^*(a,1) < T_1^*(b,1) \text{ if and only if } a < b. \tag{11.34}$$

If $C \not\supseteq (A^-)_{I_1(h,0)}$, there is $x \notin C$ such that $\mu_{A^-}(x) > I_1(h,0)$. From (11.33), we have

$$\inf_{x \notin C} I_1^*(\mu_{A^-}(x),0) < h.$$

Hence, we obtain $\pi^*_{\mathcal{A}}(C) < h$. On the other hand, if $C \not\subseteq [A^+]_{T_1(h,1)}$, there is $x \in C$ such that $\mu_{A^+}(x) < T_1(h,1)$. From (11.34), we have

$$\inf_{x \in C} T_1^*(\mu_{A^+}(x),1) < h.$$

Therefore, we obtain we obtain $\pi^*_{\mathcal{A}}(C) < h$. (Q.E.D.)

Then we have the following proposition on the calculations of possibility measures.

Proposition 11.4. *Under the assumption (A1), when $T_2(a,b) = \min(a,b)$, we have*

$$\Pi(\mathcal{B}|\mathcal{A}) = \min\left(\inf_{x \in X} I_5(\mu_{B^-}(x),\mu_{A^+}(x)), \inf_{x \in X} I_5(\mu_{A^-}(x),\mu_{B^+}(x))\right), \tag{11.35}$$

where implication function I_5 is defined by

$$I_5(a,b) = \max(I_1^{-1}(a,0), T_1^{-1}(b,1)). \tag{11.36}$$

Proof. Let $Pos(B^-,B^+;A^-,A^+)$ be the right-hand side of (11.35), that is,

$$Pos(B^-,B^+;A^-,A^+) = \min\left(\inf_{x \in X} I_5(\mu_{B^-}(x),\mu_{A^+}(x)), \inf_{x \in X} I_5(\mu_{A^-}(x),\mu_{B^+}(x))\right).$$

Suppose $Pos(B^-,B^+;A^-,A^+) \geq h$. From (11.36), this supposition is equivalent to

$$\inf_{x \in X} \max(I_1^{-1}(\mu_{B^-}(x),0), T_1^{-1}(\mu_{A^+}(x),1)) \geq h,$$
$$\inf_{x \in X} \max(I_1^{-1}(\mu_{A^-}(x),0), T_1^{-1}(\mu_{B^+}(x),1)) \geq h.$$

These can be rewritten as follows, respectively

$$\forall x \in X,\ I_1^{-1}(\mu_{B^-}(x),0) < h \text{ implies } T_1^{-1}(\mu_{A^+}(x),1) \geq h,$$
$$\forall x \in X,\ I_1^{-1}(\mu_{A^-}(x),0) < h \text{ implies } T_1^{-1}(\mu_{B^+}(x),1) \geq h.$$

Hence, we have

$$Pos(B^-,B^+;A^-,A^+) \geq h \\ \text{if and only if } (B^-)_{I_1(h,0)} \subseteq [A^+]_{T_1(h,1)} \text{ and } (A^-)_{I_1(h,0)} \subseteq [B^+]_{T_1(h,1)}. \quad (11.37)$$

On the other hand, from (11.10), we have

$$\forall h \in [0,1],\ (A^-)_{I_1(h,0)} \subseteq [A^+]_{T_1(h,1)},$$
$$\forall h \in [0,1],\ (B^-)_{I_1(h,0)} \subseteq [B^+]_{T_1(h,1)}.$$

Thus,

$$(A^-)_{I_1(h,0)} \subseteq (A^-)_{I_1(h,0)} \cup (B^-)_{I_1(h,0)} \subseteq [A^+]_{T_1(h,1)},$$
$$(B^-)_{I_1(h,0)} \subseteq (A^-)_{I_1(h,0)} \cup (B^-)_{I_1(h,0)} \subseteq [B^+]_{T_1(h,1)}.$$

From Lemma 11.1, $(A^-)_{I_1(h,0)} \cup (B^-)_{I_1(h,0)} \in [\mathcal{A}]_h \cap [\mathcal{B}]_h$. This means that

$$\Pi(\mathcal{B}|\mathcal{A}) = \sup_{C \in 2^X} \min(\pi_{\mathcal{A}}(C), \pi_{\mathcal{B}}(C)) \geq h.$$

Hence, we prove that

$$\forall h \in [0,1];\ Pos(B^-,B^+;A^-,A^+) \geq h \text{ implies } \Pi(\mathcal{B}|\mathcal{A}) \geq h,$$

in other words, $Pos(B^-,B^+;A^-,A^+) \leq \Pi(\mathcal{B}|\mathcal{A})$.

Let us prove the reverse inequality. We have

$$\Pi(\mathcal{B}|\mathcal{A}) \geq h \text{ if and only if } \forall \varepsilon > 0,\ \Pi(\mathcal{B}|\mathcal{A}) > h - \varepsilon.$$

Moreover,

$$\Pi(\mathcal{B}|\mathcal{A}) > h - \varepsilon \text{ implies } \exists C,\ C \in [\mathcal{A}]_{h-\varepsilon} \cap [\mathcal{B}]_{h-\varepsilon}.$$

From Lemma 11.1, $\exists C,\ C \in [\mathcal{A}]_{h-\varepsilon} \cap [\mathcal{B}]_{h-\varepsilon}$ means

$$(A^-)_{I_1(h-\varepsilon,0)} \subseteq C \subseteq [A^+]_{T_1(h-\varepsilon,1)} \text{ and } (B^-)_{I_1(h-\varepsilon,0)} \subseteq C \subseteq [B^+]_{T_1(h-\varepsilon,1)}.$$

By transitivity, this implies

$$(B^-)_{I_1(h-\varepsilon,0)} \subseteq [A^+]_{T_1(h-\varepsilon,1)} \text{ and } (A^-)_{I_1(h-\varepsilon,0)} \subseteq [B^+]_{T_1(h-\varepsilon,1)}.$$

From (11.37), this is equivalent to $Pos(B^-,B^+;A^-,A^+) \geq h - \varepsilon$. Therefore, we have

$$\Pi(\mathcal{B}|\mathcal{A}) \geq h \text{ implies } \forall \varepsilon > 0,\ Pos(B^-,B^+;A^-,A^+) \geq h - \varepsilon,$$

in other words, $\Pi(\mathcal{B}|\mathcal{A}) \leq Pos(B^-,B^+;A^-,A^+)$. (Q.E.D.)

Proposition 11.3 shows that, for any upper semi-continuous implication function I_2, a necessity measure of an ill-known set $\mathcal{B}$ under an ill-known set $\mathcal{A}$ is calculated by using A^-, A^+, B^-, and B^+. On the contrary, Proposition 11.4 shows that, for a special conjunction function T_2, that is, $T_2(a,b) = \min(a,b)$, possibility measure of

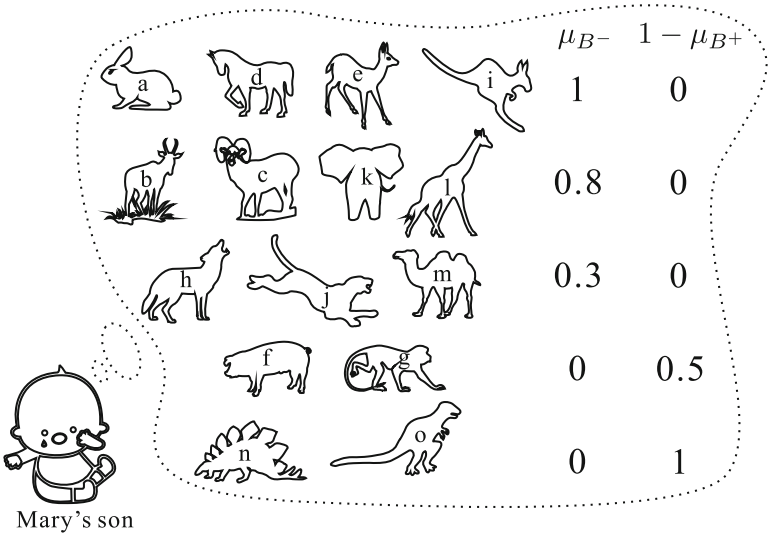


Fig. 11.5. B^- and B^+ expressing Mary's son's satisfaction degrees of sets of animals

an ill-known set $\mathcal{B}$ under an ill-known set $\mathcal{A}$ is calculated by using A^-, A^+, B^- and B^+. It does not mean that a similar result holds for other conjunction functions T_2.

Example 11.4. (a continuation of Example 2) Let $T_1(a,b) = ab$, $I_1(a,b) = 1-a+ab$, $T_2(a,b) = \min(a,b)$ and $I_2(a,b) = 1$ if $a \leq b$, $I_2(a,b) = \max(1-a,b)$ otherwise. In this case, we obtain $I_2(a,b) = I_4(a,b) = I_2(a,b)$ and $I_5(a,b) = \max(1-a,b)$. Let us discuss to what extent Mary's son is pleased with visiting the small zoo. To this end, we assume Mary's son expresses the set of animals he wants to see in the zoo by B^- and B^+ having the following membership functions (see also Figure 11.5):

$$\begin{aligned}
&\mu_{B^-}(\mathrm{a}) = 1,\ \mu_{B^+}(\mathrm{a}) = 1,\ \mu_{B^-}(\mathrm{b}) = 0.8,\ \mu_{B^+}(\mathrm{b}) = 1,\\
&\mu_{B^-}(\mathrm{c}) = 0.8,\ \mu_{B^+}(\mathrm{c}) = 1,\ \mu_{B^-}(\mathrm{d}) = 1,\ \mu_{B^+}(\mathrm{d}) = 1,\\
&\mu_{B^-}(\mathrm{e}) = 1,\ \mu_{B^+}(\mathrm{e}) = 1,\ \mu_{B^-}(\mathrm{f}) = 0,\ \mu_{B^+}(\mathrm{f}) = 0.5,\\
&\mu_{B^-}(\mathrm{g}) = 0,\ \mu_{B^+}(\mathrm{g}) = 0.5,\ \mu_{B^-}(\mathrm{h}) = 0.3,\ \mu_{B^+}(\mathrm{h}) = 1,\\
&\mu_{B^-}(\mathrm{i}) = 1,\ \mu_{B^+}(\mathrm{i}) = 1,\ \mu_{B^-}(\mathrm{j}) = 0.3,\ \mu_{B^+}(\mathrm{j}) = 1,\\
&\mu_{B^-}(\mathrm{k}) = 0.8,\ \mu_{B^+}(\mathrm{k}) = 1,\ \mu_{B^-}(\mathrm{l}) = 0.8,\ \mu_{B^+}(\mathrm{l}) = 1,\\
&\mu_{B^-}(\mathrm{m}) = 0.3,\ \mu_{B^+}(\mathrm{m}) = 1,\ \mu_{B^-}(\mathrm{n}) = 0,\ \mu_{B^+}(\mathrm{n}) = 0,\\
&\mu_{B^-}(\mathrm{o}) = 0,\ \mu_{B^+}(\mathrm{o}) = 0.
\end{aligned} \tag{11.38}$$

The necessity and possibility degrees of his satisfaction under $\pi^*_{\mathcal{A}}$ given by (11.23) are calculated as in Tables 11.1 and 11.2. Then we obtain $N(\mathcal{B}|\mathcal{A}) = 0.2$ and $\Pi(\mathcal{B}|\mathcal{A}) = 0.5$

We should be aware that the meanings of $\Pi(\mathcal{B}|\mathcal{A})$ and $N(\mathcal{B}|\mathcal{A})$ change depending on the interpretation of $\pi_{\mathcal{B}}$. For example, let us consider a situation where a manager

Table 11.1. Calculation of $N(\mathcal{B}|\mathcal{A})$

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	inf
A^+	1	1	1	1	1	1	1	1	1	1	0.5	0.5	0.5	0	0	–
B^+	1	1	1	1	1	0.5	0.5	1	1	1	1	1	1	0	0	–
I_3	1	1	1	1	1	0.5	0.5	1	1	1	1	1	1	1	1	0.5
B^-	1	0.8	0.8	1	1	0	0	0.3	1	0.3	0.8	0.8	0.3	0	0	–
A^-	1	1	1	0.5	0.5	0.5	0.5	0.25	0.25	0.25	0.25	0	0	0	0	–
I_4	1	1	1	0.5	0.5	1	1	0.7	0.25	0.7	0.25	0.2	0.7	1	1	0.2
														min		0.2

Table 11.2. Calculation of $\Pi(\mathcal{B}|\mathcal{A})$

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	inf
B^-	1	0.8	0.8	1	1	0	0	0.3	1	0.3	0.8	0.8	0.3	0	0	–
A^+	1	1	1	1	1	1	1	1	1	1	0.5	0.5	0.5	0	0	–
I_5	1	1	1	1	1	1	1	1	1	1	0.5	0.5	0.7	1	1	0.5
A^-	1	1	1	0.5	0.5	0.5	0.5	0.25	0.25	0.25	0.25	0	0	0	0	–
B^+	1	1	1	1	1	0.5	0.5	1	1	1	1	1	1	0	0	–
I_5	1	1	1	1	1	0.5	0.5	1	1	1	1	1	1	1	1	0.5
														min		0.5

wants to employ several workers. He will be able to assign his satisfaction degree to each subset of candidates. This satisfaction degree can be expressed by $\pi_{\mathcal{B}}$. In such a case, $\pi_{\mathcal{B}}$ shows the satisfaction degree of a decision maker. $\Pi(\mathcal{B}|\mathcal{A})$ and $N(\mathcal{B}|\mathcal{A})$ show the possibility and necessity degrees to what extent the decision maker is satisfied with ill-known set $\mathcal{A}$, respectively. $\Pi(\mathcal{B}|\mathcal{A})$ and $N(\mathcal{B}|\mathcal{A})$ in Example 4 are based on this interpretation.

On the other hand, if $\pi_{\mathcal{B}}(B_i)$ shows the possibility degree of the realization $B = B_i$, the meanings of $\Pi(\mathcal{B}|\mathcal{A})$ and $N(\mathcal{B}|\mathcal{A})$ are different from the previous ones. $\Pi(\mathcal{B}|\mathcal{A})$ and $N(\mathcal{B}|\mathcal{A})$ evaluate the possibility and necessity degrees to what extent A possibly equals to B.

In the latter case, we can obtain the possibility distribution of the complement of the set B from $\pi_{\mathcal{B}}$ through (11.20). Thus, we can compute $\Pi(\bar{\mathcal{B}}|\mathcal{A})$ and $N(\bar{\mathcal{B}}|\mathcal{A})$. Those measures evaluate the possibility and necessity degrees to what extent A possibly equals to $\bar{B}$. Let us introduce a strong negation $n : [0,1] \to [0,1]$ such that

(n1) $n(0) = 1$,
(n2) $n(n(a)) = a,\ a \in [0,1]$,
(n3) n is continuously decreasing.

$n(\Pi(\bar{\mathcal{B}}|\mathcal{A}))$ and $n(N(\bar{\mathcal{B}}|\mathcal{A}))$ show the impossibility and contingency degrees to what extent A possibly equals to $\bar{B}$. In other words, $n(\Pi(\bar{\mathcal{B}}|\mathcal{A}))$ and $n(N(\bar{\mathcal{B}}|\mathcal{A}))$ evaluate the necessity and possibility degrees to what extent A is certainly different from $\bar{B}$.

From Propositions 11.3 and 11.4, for $n(\Pi(\bar{\mathcal{B}}|\mathcal{A}))$ and $n(N(\bar{\mathcal{B}}|\mathcal{A}))$, we have the following corollary.

Corollary 11.1. *Under assumption (A1), the following assertions are valid:*

(a) *When I_2 is upper semi-continuous, we have*

$$n(N(\bar{\mathcal{B}}|\mathcal{A})) = \max\left(\sup_{x\in X} f_3(\mu_{A^+}(x),\mu_{B^-}(x)), \sup_{x\in X} f_4(\mu_{B^+}(x),\mu_{A^-}(x))\right), \quad (11.39)$$

where functions f_3 and f_4 are defined by

$$f_3(a,b) = n(I_2(T_1^{-1}(a,1), I_1^{-1}(b,0))), \quad (11.40)$$

$$f_4(a,b) = n(I_2(I_1^{-1}(a,0), T_1^{-1}(b,1))). \quad (11.41)$$

(b) *When $T_2(a,b) = \min(a,b)$, we have*

$$n(\Pi(\bar{\mathcal{B}}|\mathcal{A})) = \max\left(\sup_{x\in X} f_5(\mu_{B^+}(x),\mu_{A^+}(x))), \sup_{x\in X} g_5(\mu_{A^-}(x),\mu_{B^-}(x)))\right), \quad (11.42)$$

where functions f_5 and g_5 are defined by

$$f_5(a,b) = \min(n(T_1^{-1}(a,1)), n(T_1^{-1}(b,1))), \quad (11.43)$$

$$g_5(a,b) = \min(n(I_1^{-1}(a,0)), n(I_1^{-1}(b,0))). \quad (11.44)$$

Proof. From (11.27), (11.21) and (11.22), we have

$$\begin{aligned} N(\bar{\mathcal{B}}|\mathcal{A}) &= \min\left(\inf_{x\in X} I_3(\mu_{A^+}(x),\mu_{\bar{B}^+}(x)), \inf_{x\in X} I_4(\mu_{\bar{B}^-}(x),\mu_{A^-}(x))\right) \\ &= \min\left(\inf_{x\in X} I_3(\mu_{A^+}(x), T_1(I_1^{-1}(\mu_{B^-}(x),0),1)),\right. \\ &\qquad \left.\inf_{x\in X} I_4(I_1(T_1^{-1}(\mu_{B^+}(x),1),0),\mu_{A^-}(x))\right). \end{aligned}$$

Thus, we obtain

$$\begin{aligned} n(N(\bar{\mathcal{B}}|\mathcal{A})) = \max\left(\sup_{x\in X} n(I_3(\mu_{A^+}(x), T_1(I_1^{-1}(\mu_{B^-}(x),0),1))),\right. \\ \left.\sup_{x\in X} n(I_4(I_1(T_1^{-1}(\mu_{B^+}(x),1),0),\mu_{A^-}(x))\right). \end{aligned}$$

This is equivalent to (11.39).

On the other hand, from (11.35), (11.21) and (11.22), we have

$$\begin{aligned} \Pi(\bar{\mathcal{B}}|\mathcal{A}) &= \min\left(\inf_{x\in X} I_5(\mu_{\bar{B}^-}(x),\mu_{A^+}(x)), \inf_{x\in X} I_5(\mu_{A^-}(x),\mu_{\bar{B}^+}(x))\right) \\ &= \min\left(\inf_{x\in X} I_5(I_1(T_1^{-1}(\mu_{B^+}(x),1),0),\mu_{A^+}(x)),\right. \\ &\qquad \left.\inf_{x\in X} I_5(\mu_{A^-}(x), T_1(I_1^{-1}(\mu_{B^-}(x),0),1))\right). \end{aligned}$$

Thus, we obtain

$$n(\Pi(\bar{\mathcal{B}}|\mathcal{A})) = \max\left(\sup_{x\in X} n(I_5(I_1(T_1^{-1}(\mu_{B^+}(x),1),0),\mu_{A^+}(x)), \sup_{x\in X} n(I_5(\mu_{A^-}(x),T_1(I_1^{-1}(\mu_{B^-}(x),0),1))\right).$$

This is nothing but (11.42). (Q.E.D.)

As shown in Corollary 1, $n(\Pi(\bar{\mathcal{B}}|\mathcal{A}))$ and $n(N(\bar{\mathcal{B}}|\mathcal{A}))$ can also be calculated by using A^-, A^+, B^-, and B^+. Functions f_4 and g_5 are disjunction functions while functions f_3 and f_5 are negations of conjunction functions.

In the literature [14, 3], a t-norm $t : [0,1]\times[0,1] \to [0,1]$ is often used as a conjunction function, where t-norm t satisfies

(t1) $t(a,1) = t(1,a) = a$, $\forall a \in [0,1]$,
(t2) $t(a,b) \le t(c,d)$, $\forall a,b,c,d \in [0,1]$ such that $a \le c$ and $b \le d$,
(t3) $t(a,b) = t(b,a)$, $\forall a,b \in [0,1]$,
(t4) $t(a,t(b,c)) = t(t(a,b),c)$, $\forall a,b,c \in [0,1]$.

R-, S- and reciprocal R-implication functions [15, 9], that is, I^{R}, I^{S} and $I^{\mathrm{r-R}}$ are defined from a t-norm t by the following equations, respectively:

$$I^{\mathrm{R}}[t](a,b) = \sup\{s \in [0,1] \mid t(a,s) \le b\}, \tag{11.45}$$
$$I^{\mathrm{S}}[t](a,b) = n_1(t(a,n_1(b))), \tag{11.46}$$
$$I^{\mathrm{r-R}}[t](a,b) = \sup\{s \in [0,1] \mid t(n_1(b),s) \le n_1(a)\}, \tag{11.47}$$

where n_1 is a strong negation.

When T_1 in (11.13) is a t-norm and I_1 is an S- or reciprocal R-implication function, we have $T_1^{-1}(a,1) = a$ and $I_1^{-1}(a,0) = n_1(a)$, for all $a \in [0,1]$. In this case, implication functions I_3, I_4, and I_5 are expressed simply as $I_3(a,b) = I_2(a,b)$, $I_4(a,b) = I_2(n_1(b),n_1(a))$ and $I_5(a,b) = \max(n_1(a),b)$ for any $a \in [0,1]$ and $b \in [0,1]$. Moreover, f_3, f_4, f_5 and g_5 are also expressed simply as $f_3(a,b) = n(I_2(a,n_1(b)))$, $f_4(a,b) = n(I_2(n_1(a),b))$, $f_5(a,b) = \max(n(a),n(b))$ and $g_5(a,b) = \max(n(n_1(a)), n(n_1(b)))$ for any $a \in [0,1]$ and $b \in [0,1]$.

Finally, to understand differences among

$$N(\mathcal{B}|\mathcal{A}),\ \ \Pi(\mathcal{B}|\mathcal{A}),\ \ n(N(\bar{\mathcal{B}}|\mathcal{A}) \ \text{ and } n(\Pi(\bar{\mathcal{B}}|\mathcal{A})),$$

let us consider the conditions for $N(\mathcal{B}|\mathcal{A}) = 1$, $\Pi(\mathcal{B}|\mathcal{A}) = 1$, $n(N(\bar{\mathcal{B}}|\mathcal{A}) = 1$ and $n(\Pi(\bar{\mathcal{B}}|\mathcal{A})) = 1$. We assume that T_2 is a minimum operation and Assumption A1. Then, we obtain

$$N(\mathcal{B}|\mathcal{A}) = 1 \Leftarrow \mathrm{Supp}(A^+) \subseteq \mathrm{Core}(B^+) \text{ and } \mathrm{Core}(B^-) \supseteq \mathrm{Supp}(A^-), \tag{11.48}$$

$$\Pi(\mathcal{B}|\mathcal{A}) = 1 \Leftrightarrow \mathrm{Supp}(B^-) \subseteq \mathrm{Core}(A^+) \text{ and } \mathrm{Supp}(A^-) \subseteq \mathrm{Core}(B^+), \tag{11.49}$$

$$n(N(\bar{\mathcal{B}}|\mathcal{A}) = 1 \Leftarrow \mathrm{Core}(A^+) \cap \mathrm{Core}(B^-) \neq \emptyset \text{ or } \mathrm{Supp}(B^+) \cup \mathrm{Supp}(A^-) \neq X, \quad (11.50)$$

$$n(\Pi(\bar{\mathcal{B}}|\mathcal{A})) = 1 \Leftarrow \mathrm{Supp}(B^+) \cup \mathrm{Supp}(A^+) \neq X \text{ or } \mathrm{Core}(A^-) \cap \mathrm{Core}(B^-) \neq \emptyset, \quad (11.51)$$

where the inclusion relation $A \subseteq B$ between fuzzy sets A and B is defined normally by the inequality of their membership functions μ_A and μ_B, that is, $\mu_A(x) \leq \mu_B(x)$, $\forall x \in X$. We find that $N(\mathcal{B}|\mathcal{A})$ and $\Pi(\mathcal{B}|\mathcal{A})$ relate with inclusion relations among A^-, A^+, B^- and B^+ while $n(N(\bar{\mathcal{B}}|\mathcal{A})$ and $n(\Pi(\bar{\mathcal{B}}|\mathcal{A}))$ relate with intersection of B^- with A^+, A^- and intersection of complement of B^+ with complements of A^+ and A^-.

When I_2 is an R- or reciprocal R-implication, we have

$$N(\mathcal{B}|\mathcal{A}) = 1 \Leftrightarrow A^+ \subseteq B^+ \text{ and } B^- \supseteq A^-, \quad (11.52)$$

Then, under assumption A1, we find $N(\mathcal{B}|\mathcal{A}) = 1$ implies $\Pi(\mathcal{B}|\mathcal{A}) = 1$ when I_2 is an R- or reciprocal R-implication and T_2 is a minimum operation.

11.4 Concluding Remarks

In this chapter, we focused on ill-known sets imprecisely specified by their lower and upper approximations. We extended the previous results given by Dubois and Prade to cases when general implication and conjunction functions are used instead of Dienes implication and minimum operation. The simple manipulations of possibility and necessity measures of roughly defined ill-known sets have been investigated. We have introduced conditions for implication and conjunction functions to recover the original roughly defined ill-known set through the embedding and approximation processes. It has been shown that a necessity measure of ill-known set imprecisely specified by lower and upper approximations can be calculated in the universe if the necessity measure is defined by an upper semi-continuous implication function. On the other hand, we could obtain the similar result for a possibility measure of ill-known set imprecisely specified by lower and upper approximations only when the possibility measure is defined by a minimum operation. Namely, simple calculations are valid for various necessity measures and for a special possibility measure.

The specification of ill-known set is one of controversial topics. Even if we use the rough representations by lower and upper approximations, the specification of lower and upper approximations would be a complex problem. Then, the specification of ill-known set is one of future topics. Moreover, related to this, as we see in Example 2, the fusion of ill-known sets from different information sources is also an important topic. Similar to evidence theory [18], we will have various approaches.

We considered that an ill-known set is a model expressing the possible realizations of a conjunctive variable, a set-valued variable. The possible range of a disjunctive variable (i.e., a single-valued variable) can be seen as a conjunctive variable. Therefore, it can be represented by an ill-known set. In this case, we should consider a possibility distribution of possible ranges. Because possible ranges themselves include uncertainty, its possibility distribution includes further uncertainty. From this point of view, a study on ill-known sets could be valuable for treatment of higher-order uncertainty.

Finally, as an extension of ill-known sets, ill-known fuzzy sets are conceivable. In this case, we consider possibility distributions on the set of all fuzzy subsets. The manipulation of ill-known fuzzy sets on the universe using lower and upper approximations would also be an interesting topic.

Acknowledgements. The author acknowledges that this work was partially supported by the Grant-in-Aid for Scientific Research (C) No.23510169.

References

1. Atanassov, K.: lntuitionistic fuzzy sets. Fuzzy Sets and Systems 20, 87–96 (1986)
2. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. The Annals of Mathematical Statistics 38(2), 325–339 (1967)
3. Dubois, D., Prade, H.: A class of fuzzy measures based on triangular norms: A general framework for the combination of uncertain information. Int. J. Gen. Systems 8, 43–61 (1982)
4. Dubois, D., Prade, H.: Twofold fuzzy sets: an approach to the representation of sets with fuzzy boundaries based on possibility and necessity measures. Fuzzy Math. (Huazhong) 3(4), 53–76 (1983)
5. Dubois, D., Prade, H.: A set-theoretic view of belief functions: Logical connectives and approximation by fuzzy sets. Int. J. Gen. Systems 12, 193–226 (1986)
6. Dubois, D., Prade, H.: Twofold fuzzy sets and rough sets: Some issues in knowledge representation. Fuzzy Sets and Systems 23, 3–18 (1987)
7. Dubois, D., Prade, H.: Possibility Theory – An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (1988)
8. Dubois, D., Prade, H.: Incomplete conjunctive information. Comput. Math. Applic. 15, 797–810 (1988)
9. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distribution. Fuzzy Sets and Systems 40, 143–202 (1991)
10. Inuiguchi, M., Sakawa, M.: Interpretation of fuzzy reasoning based on conjunctive functions in view of uncertainty generation rules using necessity measures. Japanese Journal of Fuzzy Theory and Systems 5, 323–344 (1993)
11. Kaufmann, A.: Introduction to the Theory of Fuzzy Subsets. Academic Press, New York (1975)
12. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
13. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer, Dordrecht (1991)

14. Schweizer, B., Sklar, A.: Associative functions and statistical triangle inequalities. Publ. Math. Debrecen. 8, 169–186 (1961)
15. Trillas, E., Valverde, L.: On implication and indistinguishability in the setting of fuzzy logic. In: Kacprzyk, J., Yager, R.R. (eds.) Management Decision Support Systems Using Fuzzy Sets and Possibility Theory, pp. 198–212. Verlag TÜV Rheinland, Köln (1985)
16. Pedrycz, W.: Shadow sets: Bridging fuzzy and rough sets. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 179–199. Springer, Singapore (1999)
17. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. Inf. Sci. 34, 115–143 (1984)
18. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, New Jersey (1976)
19. Sugeno, M. (ed.): Industrial Applications of Fuzzy Control. North-Holland, New York (1985)
20. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
21. Yager, R.R.: On different classes of linguistic variables defined via fuzzy subsets. Kybernetics 13, 103–110 (1984)
22. Zadeh, L.A.: Fuzzy sets. Inform. and Control 8, 338–353 (1965)
23. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1(1), 3–28 (1978)

Chapter 12
Property-Driven Rough Sets Approximations of Relations

Ryszard Janicki*

Abstract. The problem of approximating an arbitrary relation by a relation with desired properties is formally defined and analysed. Two special cases, approximation by partial orders and approximation by equivalence relations are discussed in detail.

Keywords: Approximation of raw data, properties of relations, rough sets, α-approximations, partial order, equivalence relation.

12.1 Introduction

While, in general, sets are just arbitrary collections of arbitrary elements [8], when they are applied in other parts of Mathematics or Science, they usually have some structure and properties. Their elements are usually engaged in complex relationships. While a collection that consists of, say, a white elephant, computer mouse, empty set, and a letter 'a', is a proper set (c.f. [8, 12]), in most applications the sets are more homogenous, as 'sets of integers', 'vertices', 'variables', etc., and quite often they have some very specific structures like 'trees', 'partitions', 'partial orders', etc.

Those structures and properties are essential when it comes to the problem of *approximation* of raw empirical data by appropriate mathematical concepts.

The simplest and most abstract way of modelling complex connections relationships is to use the notion of *relation*.

Ryszard Janicki
Department of Computing and Software,
McMaster University,
Hamilton, ON, L8S 4K1 Canada
e-mail: janicki@mcmaster.ca

* Partially supported by NSERC grant of Canada.

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 333–357.
springerlink.com

The problem we will try to deal with in this chapter can be formulated as follows. We have a set of data that have been obtained in an empirical manner. From the nature of the problem, we know that the set should have some structure and desired properties, for example, it should be partially ordered, or partially ordered for one attribute and partitioned by some equivalence relation for another attribute (so it should be represented by two binary relations) but because the data are empirical it is not. In general case, this might be just an arbitrary set without the desired structure and properties. *What is the 'best' approximation that has the desired structure and properties and how it can be computed?* For the approximation of arbitrary relations by partial orders, this problem was discussed and some solutions were proposed in [6] (within both the standard theory of relations [8, 12] and Rough Sets paradigm [9, 10]).

In this chapter, we will generalise and refine some ideas of [6] to arbitrary relations, and we will illustrate our concepts by showing approximations by two of the most often used kinds of relations, partial orders and equivalence relations.

It appears that the concept of approximation has two different intuitions in Mathematics and Science. The first one stems from the fact that often, empirical numerical data have some errors, so in reality, we seldom have the value x (unless the measurements are expressible in integers) but usually some interval $(x-\varepsilon, x+\varepsilon)$, that is, the lower approximation and the upper approximation. Rough Sets [9, 10] exploit this idea for general sets. The second intuition can be illustrated by *least square approximation* of points in the two-dimensional plane (c.f. [15]). Here we know or assume that the points should be on a straight line and we are trying to find the line that fits the data best. In this case the data have a structure (points in two dimensional plane, that is, a relation that is a function) and should satisfy a desired property (be on the straight line). Note that even if we replace a solution $f(x) = ax + b$ by two lines $f_1(x) = ax + b - \delta$ and $f_2(x) = ax + b + \delta$, where δ is a standard error (c.f. [15]), there is no guarantee that any point resides between $f_1(x)$ and $f_2(x)$. Hence this is not the case of an upper, or lower approximation in the sense of Rough Sets. However this approach assumes that there is a well-defined concept of a *metric* which allows us to minimise the distance, and this concept is not obvious, and often not even possible for non-numerical objects (see for instance [6]).

The approach presented in this chapter is a mixture of both intuitions. There is no metric, but the concept of "minimal distance" is replaced and somehow simulated by a sequence of property-driven lower and/or upper approximations, in the style of Rough Sets.

The chapter is structured as follows. The next section provides basic facts about relations. Section 12.3 recalls the classical Rough Set approach to the approximation of relations. In Section 12.4, the concept of property-driven rough approximation of arbitrary relations is introduced, and the basic definitions are given. The next three sections provide basic theoretical framework for the approach presented. Section 12.5 deals with a single property (as for instance transitivity only), Section 12.6 provides an analysis of a composition of two properties (for instance symmetry and transitivity), and Section 12.7 extends the obtained results to a composition of an arbitrary number of properties. In Section 12.8, we use the ideas presented

in previous sections to approximate an arbitrary binary relation by a partial order, and in Section 12.9 to approximate an arbitrary binary relation by an equivalence relation. Section 12.8 refines some results of [6]. The last section contains final comments.

12.2 Relations and Some of Their Basic Classifications

In this section, we recall some fairly known concepts and results that will be used in the following sections [2, 8, 12].

Let X be a set, any $R \subseteq \underbrace{X \times X \times ... \times X}_{n} = \prod_{i=1}^{n} X$ is called an *n-ary relation* (on X).

If $n = 2$, that is, $R \subseteq X \times X$ then R is called a *binary relation* (on X).

Customarily, we will use the generic name *relation* for both n-ary and binary relations and apply the prefixes 'n-ary' and 'binary' only when needed. For the rest of this section, we assume that any relation is a binary relation, that is, a relation $R \subseteq X \times X$. We also will often write aRb to denote $(a,b) \in R$.

Definition 12.1 (Basic Types of Relations). Let $R, <, and \equiv$ be relations on X.

1. $id_X = \{(x,x) \mid x \in X\}$, or just id, is called the identity relation.
2. R is reflexive iff $id \subseteq R$, that is, $(x,x) \in R$ for all $x \in X$.
3. R is irreflexive iff $id \cap R = \emptyset$, that is, $(x,x) \notin R$ for all $x \in X$.
4. R is symmetric iff for all $x,y \in X$, $xRy \Rightarrow yRx$.
5. R is transitive iff for all $x,y,z \in X$, $xRy \wedge yRz \Rightarrow xRz$.
6. A relation $\equiv$ is an equivalence relation iff it is reflexive, symmetric and transitive, that is, $x \equiv x$, $x \equiv y \Rightarrow y \equiv x$, and $x \equiv y \equiv z \Rightarrow x \equiv z$, for all $x,y,z \in X$.
7. A relation $<$ is a (sharp) partial order iff it is irreflexive and transitive, that is, $\neg(x < x)$ and $x < y < z \Rightarrow x < z$ for all $x,y,z \in X$. □

For every equivalence relation $\equiv$ on X and every $x \in X$, the set $[x]_{\equiv} = \{y \mid x \equiv y\}$ denotes an *equivalence class* containing the element x.

We also have $[x]_{\equiv} = [y]_{\equiv}$ if and only if $x \equiv y$ (c.f. [2, 8, 12]).

The set of all equivalence classes of an equivalence relation $\equiv$ is denoted as $X/_{\equiv}$, and it is a *partition* of X, that is, the sets from $X/_{\equiv}$ are disjoint and cover the whole X.

For every two relations R,S on X, the relational composition $R \circ S$ is defined as $a(R \circ S)c$ if and only if $\exists b \in X.\ aRb \wedge bRc$, for all $a,c \in X$; for every relation R on X, we have $R^{-1} = \{(a,b) \mid (b,a) \in R\}$, and $R^0 = id$, $R^k = \underbrace{R \circ ... \circ R}_{k}$ for $k > 0$.

For every relation R on X, the smallest transitive (reflexive, symmetric, etc.) relation on X containing R is called the *transitive* (*reflexive*, *symmetric*, etc.) *closure* of R (c.f. [2, 12]).

Proposition 12.1 (Explicit Expresions for Closures [2, 12]). *Let R be a relation on X.*

1. $R^{ref} = R \cup id$ *is the reflexive closure of* R.
2. $R^{\overline{sym}} = R \cup R^{-1}$ *is the symmetric closure of* R.
3. $R^{+} = \bigcup_{i=1}^{\infty} R^i$ *is the transitive closure of* R.
4. $R^{*} = \bigcup_{i=0}^{\infty} R^i$ *is the reflexive-transitive closure of* R. □

Closures correspond to simple upper approximations of relations in the sense of Rough Sets. Concepts corresponding to lower approximations are more complex and less systematic and will be discussed later (in Section 12.8).

12.3 Classical Rough Relations

The principles of Rough Rets [9, 10] can be formulated as follows. Let U be a finite and nonempty universe of elements, and let $E \subseteq U \times U$ be an *equivalence relation*. The elements of $U/_E$ are called elementary sets, and they are interpreted as basic observable, measurable, or definable sets. The pair (U,E) is referred to as a Pawlak approximation space. A set $X \subseteq U$ is approximated by two subsets of U, $\underline{\mathbf{A}}(X)$ - called the lower approximation of X, and $\overline{\mathbf{A}}(X)$ - called the upper approximation of X, where $\underline{\mathbf{A}}(X)$ and $\overline{\mathbf{A}}(X)$ are defined as follows.

Definition 12.2 ([9, 10])

1. $\underline{\mathbf{A}}(X) = \bigcup\{[x]_E \mid x \in U \wedge [x]_E \subseteq X\}$,
2. $\overline{\mathbf{A}}(X) = \bigcup\{[x]_E \mid x \in U \wedge [x]_E \cap X \neq \emptyset\}$. □

Rough set approximations satisfy the following properties:

Proposition 12.2 (Pawlak [10])

1. $X \subseteq Y \Longrightarrow \underline{\mathbf{A}}(X) \subseteq \underline{\mathbf{A}}(Y)$,
2. $\underline{\mathbf{A}}(X \cap Y) = \underline{\mathbf{A}}(X) \cap \underline{\mathbf{A}}(Y)$,
3. $\underline{\mathbf{A}}(X) \subseteq X$,
4. $\underline{\mathbf{A}}(X) = \underline{\mathbf{A}}(\underline{\mathbf{A}}(X))$,
5. $\overline{\mathbf{A}}(X) = \underline{\mathbf{A}}(\overline{\mathbf{A}}(X))$,
6. $X \subseteq Y \Longrightarrow \overline{\mathbf{A}}(X) \subseteq \overline{\mathbf{A}}(Y)$,
7. $\overline{\mathbf{A}}(X \cup Y) = \overline{\mathbf{A}}(X) \cup \overline{\mathbf{A}}(Y)$,
8. $X \subseteq \overline{\mathbf{A}}(X)$,
9. $\overline{\mathbf{A}}(X) = \overline{\mathbf{A}}(\overline{\mathbf{A}}(X))$,
10. $\underline{\mathbf{A}}(X) = \overline{\mathbf{A}}(\underline{\mathbf{A}}(X))$. □

Since every relation is a set of pairs, this approach can be used for relations as well [13]. Unfortunately, in such cases as ours, we want approximations to have some specific properties like irreflexivity, transitivity etc., and most of those properties are not closed under the set union operator. As was pointed out in [17], in general, one cannot expect approximations to have the desired properties (see [17] for details). It is also unclear how to define the relation E for cases such as ours.

However, the Rough Sets can also be defined in an orthogonal (sometimes called 'topological') manner [10, 14, 16]. For a given (U,E), we may define $\mathcal{D}(U)$ as the smallest set containing $\emptyset$, all of the elements of $U/_E$ and that is closed under set union. Clearly, $U/_E$ is the set of all components generated by $\mathcal{D}(U)$ [8]. We may start with defining a space as $(U,\mathcal{D})$, where $\mathcal{D}$ is a family of sets that contains $\emptyset$, and for each $x \in U$, there is $X \in \mathcal{D}$ such that $x \in X$ (i.e. $\mathcal{D}$ is a cover of U [12]). We may now define $E_{\mathcal{D}}$ as the equivalence relation generated by the set of all components defined by $\mathcal{D}$ (see for example [8]). Hence, both approaches are equivalent [10, 14, 17]; however, now for each $X \subseteq U$, we might use different formulas for $\underline{\mathbf{A}}(X)$ and $\overline{\mathbf{A}}(X)$.

Proposition 12.3 ([10, 14, 17])

1. $\underline{\mathbf{A}}(X) = \bigcup\{Y \mid Y \subseteq X \wedge Y \in \mathcal{D}\}$,
2. $\overline{\mathbf{A}}(X) = \bigcap\{Y \mid X \subseteq Y \wedge Y \in \mathcal{D}\}$. □

We can now define $\mathcal{D}$ as a set of relations having the desired properties and then calculate $\underline{\mathbf{A}}(R)$ and/or $\overline{\mathbf{A}}(R)$ with respect to a given $\mathcal{D}$. Such an approach was proposed and analysed in [17]; however, it seems to have only limited applications. It assumes that the set $\mathcal{D}$ is closed under both union and intersection, and few properties of relations do this. For instance, transitivity is not closed under union and having a cycle is not closed under intersection. Some properties, like 'having exactly one cycle', are preserved by neither union nor intersection. This problem was discussed in [17], and they proposed that perhaps a different $\mathcal{D}$ could be used for the lower and upper approximations. But this solution again seems to have rather limited applications. The approach of [17] assumes additionally that, for the upper approximation there is at least one element of $\mathcal{D}$ that contains R, and, for the lower approximation there exists at least one element of $\mathcal{D}$ that is included in R. These are assumptions that are too strong for many applications (see [6]). If R contains a cycle, then there is no partial order that contains R!

To solve those problems, we need to create a new setting.

12.4 Property-Driven Rough Approximations of Relations

In this section, we will provide a formal basis of our approach.

Let X be a set and $\mathbf{X} = \prod_1^n X = \underbrace{X \times \ldots \times X}_{n}$. We assume that in this section any relation is an n-ary relation and a subset of $\mathbf{X}$.

Definition 12.3. 1. Any first-order predicate (c.f. [3]) α containing one atomic n-ary relational symbol R (which may occur more than once) will be called an **n-ary relational property** (or just a property).

2. Let $R \subseteq \mathbf{X}$. An n-ary relational property α is called a **property of** the n-ary relation R if the symbol R is interpreted as the relation R, all variables of α

are over the set X, and the tuple (X,R) is a model of α, that is, α holds for any assignment (c.f. [3]). □

Obvious examples of properties are *transitivity* ($\alpha = (\forall a,b,c.\ \mathsf{R}(a,b) \wedge \mathsf{R}(b,c) \Rightarrow \mathsf{R}(a,c))$), *reflexivity* ($\alpha = (\forall a.\ \mathsf{R}(a,a))$), *symmetry* ($\alpha = (\forall a,b.\ \mathsf{R}(a,b) \Rightarrow \mathsf{R}(b,a))$ etc., for binary relations. Standardly, when it does not cause any confusion, the same symbol is used to denote both R and R. We would like to point out the difference between *a property*, that is, just a statement that may or may not be true and where R is just a symbol, and *a property of* R, a statement that is true for all assignments, and R is a well-defined relation.

Definition 12.4. Let $\mathcal{P}$ be a finite set of *n-ary relational properties*, such that for each $\alpha \in \mathcal{P}$, there is a non-empty relation $Q \subseteq \mathbf{X}$, and α is a *property of* Q.

1. Any element $\alpha \in \mathcal{P}$ is called an **elementary property**.
2. For each elementary property $\alpha \in \mathcal{P}$, $P_\alpha \subseteq 2^{\mathbf{X}}$ is the set of n-ary relations over X that satisfy the property α. □

Definition 12.4 allows $\emptyset \in P_\alpha$, but disallows $P_\alpha = \emptyset$ and $P_\alpha = \{\emptyset\}$.

Even though any property can be called 'elementary', it is assumed that in any concrete case the elemetary properties are 'simple' and 'regular'. They are just atomic parts from which the real more sophisticated properties are built.

Definition 12.5. 1. For every $\alpha \in \mathcal{P}$, P_α is closed under intersection iff for each $R,S \in P_\alpha$, $R \cap S \in P_\alpha$. The set of all $\alpha \in \mathcal{P}$ that are closed under intersection will be denoted by $\mathcal{P}^{\cap}$.

2. For every $\alpha \in \mathcal{P}$, P_α is closed under union iff for each $R,S \in P_\alpha$, $R \cup S \in P_\alpha$. The set of all $\alpha \in \mathcal{P}$ that are closed under union will be denoted by $\mathcal{P}^{\cup}$. □

Some examples of properties for binary relations:

- $\alpha =$*transitivity*, or $\alpha =$*partial ordering*, P_α is closed under intersection but not under union,
- $\alpha =$*symmetry*, P_α is closed both under intersection and under union,
- $\alpha =$*having a cycle*, P_α is closed under union but not under intersection.

Assumption 1. *We assume that if* $\alpha \in \mathcal{P}$ *then,* P_α *is either closed under union or it is closed under intersection (or both), that is,* $\mathcal{P} = \mathcal{P}^{\cup} \cup \mathcal{P}^{\cap}$. □

This assumption is much weaker than it might appear as this is an assumption only about *elementary* properties, not about composite more sophisticated properties that will be considered later. However, it is absolutely needed as we want to define lower and upper approximations in the style of Proposition 12.3 and want at least one of them to exist.

Definition 12.6. The pair $(\mathbf{X}, \{P_\alpha \mid \alpha \in \mathcal{P}\})$ will be called an **property-driven approximation space for the** n**-ary relations over** X. □

When $\emptyset \in \{P_\alpha \mid \alpha \in \mathcal{P}\}$ and $\{P_\alpha \mid \alpha \in \mathcal{P}\}$ is a cover then, Definition 12.6 corresponds to the definition of space $(U, \mathcal{D})$ from Section 12.3. In Section 12.3, the elements of $\mathcal{D}$ were used to construct lower and upper approximations $\underline{\mathbf{A}}(X)$ and $\overline{\mathbf{A}}(X)$ (see Proposition 12.3), here our intention is to use the elements of $\{P_\alpha \mid \alpha \in \mathcal{P}\}$ as building bricks of our property-driven lower and upper approximations. However, as opposed to the properties of $\mathcal{D}$, it may happen that $\emptyset \notin \{P_\alpha \mid \alpha \in \mathcal{P}\}$ and that $\{P_\alpha \mid \alpha \in \mathcal{P}\}$ is not a cover.

Intuitively, for every relation R and every property $\alpha \in \mathcal{P}$, we expect an appropriate lower approximation of R to be a subset of R that belongs to P_α, and an appropriate upper approximation of R to be a superset of R that also belongs to P_α. Note that, these are weaker expectations than required from classical rough set approximations where we expect 'the largest subset of R' for lower and 'the smallest superset of R' for upper approximation. However, even this may not always be possible, which leads us to the following definition.

Definition 12.7. Let $R \subseteq \mathbf{X}$ be a non-empty relation and $\alpha \in \mathcal{P}$. We say that:

1. R has α-lower bound if and only if $\exists Q \in P_\alpha.\ Q \subseteq R$,
2. R has α-upper bound if and only if $\exists Q \in P_\alpha.\ R \subseteq Q$.

We also define

3. $lb_\alpha(R) = \{Q \mid Q \in P_\alpha \wedge Q \subseteq R\}$, the set of all α-lower bounds of R, and
4. $ub_\alpha(R) = \{Q \mid Q \in P_\alpha \wedge R \subseteq Q\}$, the set of all α-upper bounds of R. □

Note that, if the relation $\mathbf{X} = \prod_{i=1}^{n} X$ satisfies α, then α-upper bound exists for any $R \subseteq \mathbf{X}$, and if the relation $\emptyset$ satisfies α, then α-lower bound exists for any $R \subseteq \mathbf{X}$.

Some examples for binary relations:

- α =*transitivity*, R - any relation, both α-lower bound and α-upper bound do exist,
- α =*reflexivity*, R - any relation, α-lower bound exists only when R is already reflexive, α-upper bound does exist,
- α =*irreflexivity*, R - any relation, α-lower bound does exist, α-upper bound exists only when R is already irreflexive,
- α =*symmetry*, R - any relation, α-lower bound may not exist while α-upper bound does exist,
- α =*partial ordering*, R has a cycle, α-lower bound exists but α-upper bound does not exist.
- α =*partial ordering*, R - any relation, α-lower bound exists, α-upper bound may not exist,
- α =*equivalence*, R - any relation, α-lower bound exists only when R is reflexive, α-upper bound does exist,
- α =*having a cycle*, R is a partial order, α-lower bound does not exist, α-upper bound exists,

- α =*having a cycle*, R - any relation, α-lower bound may not exist, α-upper bound exists,
- $\alpha = (R(a,b) \wedge \neg R(c,d))$, R any relation such that $(a,b) \notin R$ and $(c,d) \in R$, neither α-lower bound nor α-upper bound exists.

The remaining auxilliary concepts that are needed to formally define lower and upper approximations that preserve elementary properties of relations are the well known concepts of maximal and minimal elements of families of relations (c.f. [2, 12]).

Definition 12.8. For every family of relations $\mathcal{F} \subseteq 2^{\mathbf{X}}$, we define

1. $min(\mathcal{F}) = \{R \mid \forall Q \in \mathcal{F}.\, Q \subseteq R \Rightarrow R = Q\}$, the set of all minimal elements of $\mathcal{F}$,
2. $max(\mathcal{F}) = \{R \mid \forall Q \in \mathcal{F}.\, R \subseteq Q \Rightarrow R = Q\}$, the set of all maximal elements of $\mathcal{F}$.

□

We are now able to provide the two main definitions of our model.

Definition 12.9 (α-lower and α-upper approximations)

1. If R has α-lower bound then we define its α-*lower approximation* as:

$$\underline{\mathbf{A}}_\alpha(R) = \bigcap \{Q \mid Q \in max(lb_\alpha(R))\}.$$

2. If R has α-upper bound then we define its α-*upper approximation* as:

$$\overline{\mathbf{A}}_\alpha(R) = \bigcup \{Q \mid Q \in min(ub_\alpha(R))\}.$$ □

If R *does not have* α-lower bound (α-upper bound) then its α-lower approximation (α-upper approximation) *does not exist*. This is the major difference between this model and the standard Rough Sets model. It might happen that neither α-lower approximation nor α-upper approximation exists. Then α should probably not be called an 'elementary' property and it should instead be decomposed into a conjunction of simpler properties. This problem will be discussed in Section 12.6.

To show that Definition 12.9 is sound, we need to prove the following:

(1) if the relation R has a property α, both approximations are reduced to identity,
(2) for every property $\alpha \in \mathcal{P}$, both $\underline{\mathbf{A}}_\alpha(R)$ and $\overline{\mathbf{A}}_\alpha(R)$ satisfy the property α (if they exist), first of all this is what they were invented for,
(3) when a property α is closed under both union and intersection, and an α-(lower/upper) approximation exists, it should be identical to the standard lower/upper approximation (either that of Definition 12.2 or its equivalent version from Proposition 12.3).

The result below proves the point (1).

Proposition 12.4. *If* $R \in P_\alpha$ *then* $\underline{\mathbf{A}}_\alpha(R) = \overline{\mathbf{A}}_\alpha(R) = R$.

Proof. If $R \in P_\alpha$ then $lb_\alpha(R) = ub_\alpha(R) = \{R\}$. □

The proof of point (2) will be split into two parts.

Proposition 12.5

1. *If* $\alpha \in \mathcal{P}^\cap$ *and R has* α*-lower bound then*
$$\underline{\mathbf{A}}_\alpha(R) = \bigcap\{Q \mid Q \in max(lb_\alpha(R))\} \in P_\alpha.$$
2. *If* $\alpha \in \mathcal{P}^\cup$ *and R has* α*-upper bound then*
$$\overline{\mathbf{A}}_\alpha(R) = \bigcup\{Q \mid Q \in min(ub_\alpha(R))\} \in P_\alpha.$$

Proof
(1) Every element of $max(lb_\alpha(R))$ is in P_α. Since $\alpha \in \mathcal{P}^\cap$, the intersection of all elements of $max(lb_\alpha(R))$ is also in P_α.
(2) Every element of $min(ub_\alpha(R))$ is in P_α. Since $\alpha \in \mathcal{P}^\cup$, the union of all elements of $min(ub_\alpha(R))$ also is in P_α. □

The second part involves new representations of both $\underline{\mathbf{A}}_\alpha(R)$ and $\overline{\mathbf{A}}_\alpha(R)$, more or less in the style of $\underline{\mathbf{A}}(R)$ and $\overline{\mathbf{A}}(R)$ from Proposition 12.3.

Proposition 12.6

1. *If* $\alpha \in \mathcal{P}^\cup$ *and R has* α*-lower bound, then*
$$\underline{\mathbf{A}}_\alpha(R) = \bigcup\{Q \mid Q \in lb_\alpha(R)\} = \bigcup\{Q \mid Q \subseteq R \wedge Q \in P_\alpha\} \in P_\alpha.$$
2. *If* $\alpha \in \mathcal{P}^\cap$ *and R has* α*-upper bound, then*
$$\overline{\mathbf{A}}_\alpha(R) = \bigcap\{Q \mid Q \in ub_\alpha(R)\} = \bigcap\{Q \mid R \subseteq Q \wedge Q \in P_\alpha\} \in P_\alpha.$$

Proof
(1) If $\alpha \in \mathcal{P}^\cup$ and R has α-lower bound, then $max(lb_\alpha(R))$ is a singleton set, that is, $max(lb_\alpha(R)) = \{Q\}$, where $Q = \bigcup\{S \mid S \in lb_\alpha(R)\}$. Every element of $lb_\alpha(R)$ is clearly in P_α. Since $\alpha \in \mathcal{P}^\cup$, the union of all elements of $lb_\alpha(R)$ is in P_α, that is, $Q \in P_\alpha$.
(2) If $\alpha \in \mathcal{P}^\cap$ and R has α-upper bound, then $min(ub_\alpha(R))$ is a singleton set, that is, $min(ub_\alpha(R)) = \{S\}$, where $S = \bigcap\{Q \mid Q \in ub_\alpha(R)\}$. Since $\alpha \in \mathcal{P}^\cap$, the intersection of all elements of $ub_\alpha(R)$ is in P_α, that is, $S \in P_\alpha$. □

The next result shows *when this model is exactly the same as the classical Rough Sets approach to relations* (the version from [16, 17] illustrated by Proposition 12.3). It is a proof of point (3) of the soudness requirements.

Corollary 12.1

1. *If* $\alpha \in \mathcal{P}^\cup \cap \mathcal{P}^\cap$ *and R has* α*-lower bound, then* $\underline{\mathbf{A}}_\alpha(R) = \underline{\mathbf{A}}(R)$*, and,*
2. *if* $\alpha \in \mathcal{P}^\cup \cap \mathcal{P}^\cap$ *and R has* α*-upper bound, then* $\overline{\mathbf{A}}_\alpha(R) = \overline{\mathbf{A}}(R)$*,*

where $\underline{\mathbf{A}}(R)$ *and* $\overline{\mathbf{A}}(R)$ *are classical upper and lower rough approximations over the space* $(X \times X, \{P_\alpha \mid \alpha \in \mathcal{P}\})$*, as defined in Proposition 12.3.*

Proof (1) From the second equality in Proposition 12.6(1).
(2) From the second equality in Proposition 12.6(2). □

In this section, we defined lower and upper approximations that provide desired relational properties. In the next section, we will discuss major properties of these approximations.

12.5 Properties of α-Approximations

In this section, we will show that the operational and compositional properties of α-lower and α-upper approximations are pretty close (but not identical) to those of standard rough set approximations as presented in Proposition 12.2. We start with the properties of α-lower approximation (compare with Proposition 12.2(1–4) for standard rough set lower approximation).

Proposition 12.7. *If* $R, Q \subseteq \mathbf{X}$ *have* α*-lower bound then:*

1. $R \subseteq Q \Longrightarrow \underline{\mathbf{A}}_\alpha(R) \subseteq \underline{\mathbf{A}}_\alpha(Q)$,
2. $\underline{\mathbf{A}}_\alpha(R) \subseteq R$,
3. $\underline{\mathbf{A}}_\alpha(R) = \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R))$,
4. $\underline{\mathbf{A}}_\alpha(R \cap Q) = \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q))$,
5. *if* $\alpha \in \mathcal{P}^{\cap}$ *then* $\underline{\mathbf{A}}_\alpha(R \cap Q) = \underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q)$,
6. *if* R *has* α*-upper bound then* $\overline{\mathbf{A}}_\alpha(R) = \underline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(R))$.

Proof.
(1) Since $R \subseteq Q \Longrightarrow lb_\alpha(R) \subseteq lb_\alpha(Q) \Longrightarrow max(lb_\alpha(R)) \subseteq lb_\alpha(Q)$, then for each $S \in max(lb_\alpha(R))$, there is $S' \in max(lb_\alpha(Q))$ such that $S \subseteq S'$; and intersection preserves inclusion.
(2) Since $S \in lb_\alpha(R) \Longrightarrow S \subseteq R$, and intersection preserves inclusion.
(3) From Proposition 12.4 because $\underline{\mathbf{A}}_\alpha(R) \in P_\alpha$.
(4) By (1) we have $\underline{\mathbf{A}}_\alpha(R \cap Q) \subseteq \underline{\mathbf{A}}_\alpha(R)$ and $\underline{\mathbf{A}}_\alpha(R \cap Q) \subseteq \underline{\mathbf{A}}_\alpha(Q)$, so $\underline{\mathbf{A}}_\alpha(R \cap Q) \subseteq \underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q)$. Hence, by (2) and (3) $\underline{\mathbf{A}}_\alpha(R \cap Q) \subseteq \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q))$.
By the definition, we have $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q)) = \bigcap\{S \mid S \in max(lb_\alpha(\underline{\mathbf{A}}_\alpha(Q) \cap \underline{\mathbf{A}}_\alpha(Q)))\}$. Let $T \in lb_\alpha(\underline{\mathbf{A}}_\alpha(Q) \cap \underline{\mathbf{A}}_\alpha(Q)))$. This means $T \in P_\alpha \wedge T \subseteq \underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q)$; hence, $T \in P_\alpha \wedge T \subseteq R \wedge T \subseteq Q$, that is, $T \in P_\alpha \wedge T \subseteq R \cap Q$. Therefore, $T \in lb_\alpha(R \cap Q)$. In this way, we proved that $lb_\alpha(\underline{\mathbf{A}}_\alpha(Q) \cap \underline{\mathbf{A}}_\alpha(Q)) \subseteq lb_\alpha(R \cap Q)$. Hence,
$max(lb_\alpha(\underline{\mathbf{A}}_\alpha(Q) \cap \underline{\mathbf{A}}_\alpha(Q))) \subseteq lb_\alpha(X \cap Q)$, that is, for each $S \in max(lb_\alpha(\underline{\mathbf{A}}_\alpha(Q) \cap \underline{\mathbf{A}}_\alpha(Y)))$, there exists $S' \in max(lb_\alpha(R \cap Q))$, such that $S \subseteq S'$. Since intersection preserves inclusion, this means that $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q)) \subseteq \underline{\mathbf{A}}_\alpha(R \cap Q)$.
(5) By (4) of this proposition, we have $\underline{\mathbf{A}}_\alpha(R \cap Q) = \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q))$. If $\alpha \in \mathcal{P}^{\cap}$ then, $\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q) \in P_\alpha$, so by Proposition 12.4, we have
$\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q) = \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cap \underline{\mathbf{A}}_\alpha(Q))$.

(6) If R has α-upper bound then $\overline{\mathbf{A}}_\alpha(R) \in P_\alpha$ so from Proposition 12.4, it follows that,
$\overline{\mathbf{A}}_\alpha(X) = \underline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(X))$. □

The difference from the classical case is that intersection splits into two cases and mixing lower with upper α-approximation is conditional.

We will now present the properties of α-upper approximation (compare with Proposition 12.2(5–10) for standard rough set upper approximation).

Proposition 12.8. *If $R, Q \subseteq \mathbf{X}$ have α-upper bound then*

1. $R \subseteq Q \Longrightarrow \overline{\mathbf{A}}_\alpha(R) \subseteq \overline{\mathbf{A}}_\alpha(Q)$,
2. $R \subseteq \overline{\mathbf{A}}_\alpha(R)$,
3. $\overline{\mathbf{A}}_\alpha(R) = \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(R))$,
4. $\overline{\mathbf{A}}_\alpha(R \cup Q) = \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q))$,
5. *if* $\alpha \in \mathcal{P}^{\cup}$ *then* $\overline{\mathbf{A}}_\alpha(R \cup Q) = \overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q)$,
6. *if R has α-lower bound then* $\underline{\mathbf{A}}_\alpha(R) = \overline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R))$.

Proof.
(1) Since $R \subseteq Q \Longrightarrow ub_\alpha(Q) \subseteq ub_\alpha(R) \Longrightarrow min(ub_\alpha(Q)) \subseteq ub_\alpha(R)$, then for each $S' \in min(ub_\alpha(Q))$ there is $S \in min(ub_\alpha(R))$ such that $S \subseteq S'$; and union preserves inclusion.
(2) Since $S \in ub_\alpha(R) \Longrightarrow R \subseteq S$, and union preserves inclusion.
(3) From Proposition 12.4 because $\overline{\mathbf{A}}_\alpha(R) \in P_\alpha$.
(4) By (1) we have $\overline{\mathbf{A}}_\alpha(R) \subseteq \overline{\mathbf{A}}_\alpha(R \cup Q)$ and $\overline{\mathbf{A}}_\alpha(Q) \subseteq \underline{\mathbf{A}}_\alpha(R \cup Q)$, so $\overline{\mathbf{A}}_\alpha(R) \cup \underline{\mathbf{A}}_\alpha(Q) \subseteq \underline{\mathbf{A}}_\alpha(R \cup Q)$. Hence, by (2) and (3) $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(R) \cup \underline{\mathbf{A}}_\alpha(Q)) \subseteq \overline{\mathbf{A}}_\alpha(R \cup Q)$. Since $R \subseteq \overline{\mathbf{A}}_\alpha(R)$ and $Q \subseteq \overline{\mathbf{A}}_\alpha(Q)$ then, $R \cup Q \subseteq \overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q)$, i.e. $up_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q) \subseteq up_\alpha(R \cup Q)$, and consequently, $min(up_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q)) \subseteq up_\alpha(R \cup Q)$. Hence, for each $S' \in min(up_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q))$, there exists $S \in min(up_\alpha(R \cup Q))$ such that $S \subseteq S'$. Since union preserves inclusion, we obtained $\overline{\mathbf{A}}_\alpha(R \cup Q) \subseteq \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q))$.
(5) By (4) of this proposition, $\overline{\mathbf{A}}_\alpha(R \cup Q) = \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q))$. If $\alpha \in \mathcal{P}^{\cup}$ then $\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q) \in P_\alpha$, so by Proposition 12.4 we have $\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q) = \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\alpha(R) \cup \overline{\mathbf{A}}_\alpha(Q))$.
(6) If R has α-lower bound then, $\underline{\mathbf{A}}_\alpha(R) \in P_\alpha$ so by Proposition 12.4 we have, $\underline{\mathbf{A}}_\alpha(X) = \overline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\alpha(X))$. □

Here the difference from the classical case is that union splits into two cases and mixing upper with lower α-approximation is conditional.

12.6 Composite Properties

Most of the interesting properties are composite properties. For example, a binary relation can be made a partial order by applying transitive closure first and making the outcome acyclic later, or in the opposite order (see Section 12.8 and [6]), or a

relation can be made an equivalence relation by applying relexive, symmetric and transitive closures in this order (see Section 12.9). In this section, we will propose a framework for doing this kind of compositions in a systematic way.

In principle, we will try to solve the following problem. Suppose we have two properties α and β, but we are really interested in the property $\alpha \wedge \beta$. Under what circumstances do the approximations $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R))$, $\overline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R))$, $\underline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R))$ and $\overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R))$ exist and satisfy the property $\alpha \wedge \beta$? What is the relationship between $\underline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R))$ and $\overline{\mathbf{A}}_\beta(\underline{\mathbf{A}}_\alpha(R))$, $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R))$ and $\underline{\mathbf{A}}_\beta(\underline{\mathbf{A}}_\alpha(R))$, etc.? What about the relationship between the approximations $\underline{\mathbf{A}}_{\alpha\wedge\beta}(R)$ and $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R))$, and between $\overline{\mathbf{A}}_{\alpha\wedge\beta}(R)$ and $\overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R))$?

We will restrict our attention to the conjuction operator '$\wedge$' only. The other two basic operators of propositional logic, conjuction '$\vee$', and negation '$\neg$', will not be discussed. Adding them to this model is an open research problem. However, most of the popular properties in Science and also in Mathematics are defined as conjuctions of two or more basic properties.

Hence we start with the following definition.

Definition 12.10. $\mathcal{P}^\wedge = \{\alpha_1 \wedge \ldots \wedge \alpha_k \mid k \geq 1 \text{ and } \alpha_i \in \mathcal{P} \text{ for } i = 1,\ldots,k\}$.
The elements of $\mathcal{P}^\wedge$ are called **composite properties**. □

Propositions 12.5 and 12.6 guarantee that if either $\underline{\mathbf{A}}_\alpha(R)$ or $\overline{\mathbf{A}}_\alpha(R)$ exists, it satisfies the property α. But if R has a property β different from α, neither $\underline{\mathbf{A}}_\alpha(R)$ nor $\overline{\mathbf{A}}_\alpha(R)$ may satisfy β. For example if R is transitive, its symmetric closure is symmetric, but may not be transitive any longer [12].

Definition 12.11. Let $\alpha, \beta \in \mathcal{P}^\wedge$.

1. We say that α **l-preserves** β iff
   for every $R \in P_\beta$, if R has α-lower bound then $\underline{\mathbf{A}}_\alpha(R) \in P_\beta$,
2. We say that α **u-preserves** β iff
   if R has α-upper bound then $\overline{\mathbf{A}}_\alpha(R) \in P_\beta$. □

The result below validates the above definition.

Proposition 12.9. *Let $\alpha, \beta \in \mathcal{P}^\wedge$. Then we have:*

1. *if α l-preserves β, R has β-lower bound and $\underline{\mathbf{A}}_\beta(R)$ has α-lower bound, then $S = \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R)) \in P_\alpha \cap P_\beta$, that is, S satisfies the property $\alpha \wedge \beta$.*
2. *if α l-preserves β, R has β-upper bound and $\overline{\mathbf{A}}_\beta(R)$ has α-lower bound, then $S = \underline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R)) \in P_\alpha \cap P_\beta$, that is, S satisfies the property $\alpha \wedge \beta$.*
3. *if α u-preserves β, R has β-lower bound and $\underline{\mathbf{A}}_\beta(R)$ has α-upper bound, then $S = \overline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R)) \in P_\alpha \cap P_\beta$, that is, S satisfies the property $\alpha \wedge \beta$.*
4. *if α u-preserves β, R has β-upper bound and $\overline{\mathbf{A}}_\beta(R)$ has α-upper bound, then $S = \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R)) \in P_\alpha \cap P_\beta$, that is, S satisfies the property $\alpha \wedge \beta$.*

Proof. (1) Since R has β-lower bound, by Proposition 12.5(1) - if $\beta \in \mathcal{P}^{\cap}$, or Proposition 12.6(1) - if $\beta \in \mathcal{P}^{\cup}$, then $Q = \underline{\mathbf{A}}_{\beta}(R) \in P_{\beta}$. Since α l-preserves β and $Q = \underline{\mathbf{A}}_{\beta}(R)$ has α-lower bound, then by Definition 12.11(1), $S = \underline{\mathbf{A}}_{\alpha}(Q) \in P_{\beta}$. Again, since $Q = \underline{\mathbf{A}}_{\beta}(R)$ has α-lower bound, by Proposition 12.5(1) - if $\alpha \in \mathcal{P}^{\cap}$, or Proposition 12.6(1) - if $\alpha \in \mathcal{P}^{\cup}$, then $S = \underline{\mathbf{A}}_{\alpha}(Q) \in P_{\alpha}$. Hence $S = \underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \in P_{\alpha} \cap P_{\beta}$. (2), (3) and (4) are carried out similarly as (1). □

In general there are no specific relationships between $\underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R))$ and $\underline{\mathbf{A}}_{\beta}(\underline{\mathbf{A}}_{\alpha}(R))$, or between $\overline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R))$ and $\overline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R))$. The approximation $\underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R))$ may exist but $\underline{\mathbf{A}}_{\beta}(\underline{\mathbf{A}}_{\alpha}(R))$ may not, and similarly for upper approximations. Even if they both exist, they may be equal, not equal, one included into another - or not, etc., some examples will be discussed in Sections 12.8 and 12.9. However, there is a very specific relationship between $\underline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R))$ and $\overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R))$.

Proposition 12.10. *Let $\alpha, \beta \in \mathcal{P}^{\wedge}$, and*

- *α u-preserves β and β l-preserves α,*
- *R has α-upper bound and β-lower bound,*
- *$\overline{\mathbf{A}}_{\alpha}(R)$ has β-lower bound, and*
- *$\underline{\mathbf{A}}_{\beta}(R)$ has α-lower bound*

then

1. $\overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \in P_{\alpha} \cap P_{\beta}$ *and* $\underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R)) \in P_{\alpha} \cap P_{\beta}$.
2. $\overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \subseteq \underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R))$.

Proof. (1) By Proposition 12.9(2) and 12.9(3).
(2) By Proposition 12.8(2), $R \subseteq \overline{\mathbf{A}}_{\alpha}(R)$, so $\underline{\mathbf{A}}_{\beta}(R) \subseteq \underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R))$, and $\overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \subseteq \overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R)))$. By (1) of this proposition, $\underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R)) \in P_{\alpha}$, so by Proposition 12.4, $\overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R))) = \underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R))$. Therefore $\overline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \subseteq \underline{\mathbf{A}}_{\beta}(\overline{\mathbf{A}}_{\alpha}(R))$. □

Note that, the above result is consistent with Propositions 12.7(6) and 12.8(6). We would like to point out that, in general, there is *no* inclusion-type relationship between R and $\underline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R))$ and similary for R and $\overline{\mathbf{A}}_{\beta}(\underline{\mathbf{A}}_{\alpha}(R)$.

Suppose that both $\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$ and $\underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R))$ (or $\overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$ and $\overline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R))$) exist and $\underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \in P_{(\alpha\wedge\beta)}$ (or $\overline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R)) \in P_{(\alpha\wedge\beta)}$). Which one is a better approximation of R?

Proposition 12.11. *Assume that α, β belong to $\mathcal{P}^{\wedge}$.*

1. *If R has β-lower bound and $(\alpha \wedge \beta)$-lower bound, and $\underline{\mathbf{A}}_{\beta}(R)$ has α-lower bound, then:* $\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R) \subseteq \underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R)) \subseteq R$,
2. *If R has β-upper bound and $(\alpha \wedge \beta)$-upper bound, and $\overline{\mathbf{A}}_{\beta}(R)$ has α-upper bound, then:* $R \subseteq \overline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R)) \subseteq \overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$.
3. $\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$, $\overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$, $\underline{\mathbf{A}}_{\alpha}(\underline{\mathbf{A}}_{\beta}(R))$, $\overline{\mathbf{A}}_{\alpha}(\overline{\mathbf{A}}_{\beta}(R)) \in P_{(\alpha\wedge\beta)} = P_{\alpha} \cap P_{\beta}$.

Proof. (1) Since obviously $lb_{(\alpha\wedge\beta)}(R) \subseteq lb_\beta(R)$ then $\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R) \subseteq \underline{\mathbf{A}}_\beta(R)$. Hence $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)) \subseteq \underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R))$. Since $\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R) \in P_\alpha$, then due to Proposition 12.4, $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)) = \underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$. From Proposition 12.7(2) we have $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R)) \subseteq R$, which ends the proof of (1).
(2) Since obviously $ub_{(\alpha\wedge\beta)}(R) \subseteq ub_\beta(R)$ then $min(ub_{(\alpha\wedge\beta)}(R)) \subseteq ub_\beta(R)$. This means $\overline{\mathbf{A}}_\beta(R) \subseteq \overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$. Hence $\overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R)) \subseteq \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R))$. Since $\overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R) \in P_\alpha$, then due to Proposition 12.4, $\overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)) = \overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$. From Proposition 12.8(2) we have $R \subseteq \overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R))$, which ends the proof of (2).
(3) $\underline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$ and $\overline{\mathbf{A}}_{(\alpha\wedge\beta)}(R)$ belong to $P_{(\alpha\wedge\beta)}$ by either Proposition 12.5 or Proposition 12.6. $\underline{\mathbf{A}}_\alpha(\underline{\mathbf{A}}_\beta(R)) \in P_\alpha \cap P_\beta$ by Proposition 12.9(1), and $\overline{\mathbf{A}}_\alpha(\overline{\mathbf{A}}_\beta(R)) \in P_\alpha \cap P_\beta$ by Proposition 12.9(4). □

Proposition 12.11 suggests an important technique for the design of approximation schema. It in principle says that using a complex predicate as a property usually results in a *worse* approximation than when the property is decomposed into simpler ones, and then we approximate a given relation over all these simpler properties. This means that before starting an approximation process we should think carefully how the given property could be decomposed into the simpler (and more regular with respect to the theory presented) ones.

12.7 Mixed Approximations

Many properties may have the form like $\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_k$, and in general only some sequences of α_i-approximations could lead to the desired result. In this section we will provide some framework to deal with this problem.

We adopt the following convention, we will often write $\mathbf{A}^{(0)}_\alpha(R)$ instead of $\underline{\mathbf{A}}_\alpha(R)$ and $\mathbf{A}^{(1)}_\alpha(R)$ instead of $\overline{\mathbf{A}}_\alpha(R)$.

Definition 12.12. A sequence $s = (\alpha_1, i_1)(\alpha_2, i_2)...(\alpha_k, i_k)$, where $k \geq 1$, $\alpha_j \in \mathcal{P}$ and $i_j \in \{0,1\}$ for $j = 1,...,k$, is a **proper approximation schedule** of a given relation $R \subseteq \mathbf{X}$, iff the following conditions are satisfied

1. $\alpha_i \neq \alpha_{i+1}$, for $i = 1,...,k-1$, and
2. the **mixed approximation** $\mathbf{A}^s(R)$, defined as

$$\mathbf{A}^s(R) = \mathbf{A}^{(i_1)}_{\alpha_1}(\mathbf{A}^{(i_2)}_{\alpha_2}(...(\mathbf{A}^{(i_k)}_{\alpha_k}(R))...))$$

does exist and $\mathbf{A}^s(R) \in \mathcal{P}_{(\alpha_1\wedge...\wedge\alpha_k)}$.

A conjunction $\pi(s) = \alpha_1 \wedge \alpha_2 \wedge .... \wedge \alpha_k$ is the **composite property generated by** the sequence s. □

We will also write $\alpha^{(0)}$ instead of $(\alpha,0)$, $\alpha^{(1)}$ instead of $(\alpha,1)$, 'α-0 bound' instead of 'α-lower bound', 'α-1 bound' instead of 'α-upper bound', '0-preserves' instead of 'l-preserves', and '1-preserves' instead of 'u-preserves'.

Proposition 12.12. *Let $R \subseteq \mathbf{X}$ and $s = \alpha_1^{(i_1)}\alpha_2^{(i_2)}....\alpha_k^{(i_k)}$ be a sequence with $\alpha_i \neq \alpha_{i+1}$, for $i = 1,...,k-1$. Define subsequences of s as follows: $s_k = \alpha_k^{(i_k)}$, $s_{k-1} = \alpha_{k-1}^{(i_{k-1})}s_k$,..., $s_2 = \alpha_2^{(i_2)}s_3$, $s_1 = \alpha_1^{(i_1)}s_2 = s$.*
The sequence $s = \alpha_1^{(i_1)}\alpha_2^{(i_2)}....\alpha_k^{(i_k)}$ is a proper approximation schedule of the relation R iff the following conditions are satisfied:

1. *R has α_k-i_k bound,*
2. *for each $j = k-1,...,1$, α_j i_j-preserves $\pi(s_{j+1})$ and $\mathbf{A}^{s_{j+1}}(R)$ has α_j-i_j bound.*

Proof. By induction on the length of s, using Propositions 12.3, 12.6 and 12.9. □

While Definition 12.12 is not constructive, Proposition 12.12 suggests a recursive algorithm that can be used to compute $\mathbf{A}^s(R)$.

12.8 Approximations by Partial Orders

The theory proposed above can be applied to any composite property and any relation. We will now apply it to the approximation of arbitrary binary relations by partial orders. This section is a refined version of the results initially presented in [6].

We start with defining two operations on binary relations that will later be used to construct partial order approximations.

Let X be a set and $R \subseteq X \times X$ be any relation.

Definition 12.13 ([6]). Let $R \subseteq X \times X$.

1. The relation $R^\bullet$, the **acyclic refinement** of R, is defined as follows:

$$aR^\bullet b \text{ if and only if } aRb \wedge \neg(aR^{cyc}b),$$

   where $aR^{cyc}b$if and only if $aR^+b \wedge bR^+a$, that is, $aR^{cyc}b$ means a and b belong to some cycle.
2. The relation $R^\subset$, the **inclusion property kernel** of R, is defined as:

$$aR^\subset b \text{ if and only if } bR^{ref} \subset aR^{ref} \wedge R^{ref}a \subset R^{ref}b,$$

   where R^{ref} is a reflexive closure of R, and for every relation $S \subseteq X \times X$,
   $Sa = \{x \mid xSa\}$ and $aS = \{x \mid aSx\}$.
3. The relation $R^{\subset\wedge\bullet}$ is defined as follows: $aR^{\subset\wedge\bullet}b$if and only if $aR^\subset b \wedge aR^\bullet b$. □

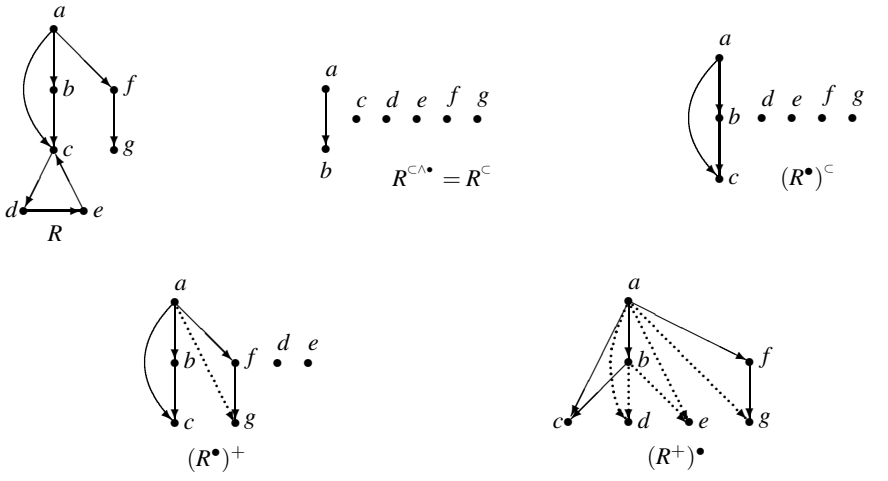


Fig. 12.1. [6] An example of a relation R, its partial order approximations $R^{\subset\wedge\bullet}$, $R^{\subset}$, $(R^{\bullet})^{\subset}$, $(R^{\bullet})^{+}$ and $(R^{+})^{\bullet}$. Dotted lines in $(R^{\bullet})^{+}$ and $(R^{+})^{\bullet}$ indicate the relationship that is not in R and was added by the transitivity operation. In general, we only have $R^{\subset\wedge\bullet} \subseteq R^{\subset}$ and it might happen that $R^{\subset\wedge\bullet} \neq R^{\subset}$ (c.f. [6]).

The word 'kernel' is often used as an antonym of 'closure'. While 'closure' is defined as the least superset having a desired property, the word 'kernel' is often used to name the greatest subset having a desired property. The 'inclusion property kernel' is a kernel in this sense. While $R^{\bullet}$ is an acyclic subset of R, it is not a kernel, as the greatest acyclic subrelation usually does not exist. Hence, the name 'refinement' was proposed and used (see [6]).

Theorem 12.1 ([6]). *For every relation $R \subseteq X \times X$, we have.*

1. *The relations $R^{\subset\wedge\bullet}$, $R^{\subset}$, $(R^{\bullet})^{\subset}$, $(R^{\bullet})^{+}$, and $(R^{+})^{\bullet}$ are partial orders, and can be considered as partial order approximations of R.*
2. $R^{\subset\wedge\bullet} \subseteq (R^{\bullet})^{\subset} \subseteq (R^{\bullet})^{+} \subseteq (R^{+})^{\bullet}$.
3. $R^{\subset\wedge\bullet} \subseteq (R^{\subset})^{\bullet} = R^{\subset} \subseteq R$. □

The statements like $(R^{\bullet})^{\subset}$ should be read as follows, find the acyclic refinement of R first and then find the inclusion property kernel of $R^{\bullet}$.

The relations $R^{\subset\wedge\bullet}$, $R^{\subset}$, $(R^{\bullet})^{\subset}$, $(R^{\bullet})^{+}$, $(R^{+})^{\bullet}$ and Theorem 12.1 are illustrated by an example in Figure 12.1 (from [6]). A formal definition of a partial order approximation of arbitrary binary relations has been given and justified in [6] and all the relations $R^{\subset\wedge\bullet}$, $R^{\subset}$, $(R^{\bullet})^{\subset}$, $(R^{\bullet})^{+}$, $(R^{+})^{\bullet}$ satisfy that definition. The idea of using the relation $(R^{+})^{\bullet}$ as a partial order approximation came from Schröder (1985) and was initially formulated in terms of quasi-orders (see [6]). The other approximations, to our knowledge, originated from [6] and its conference predecessor. There is no universal inclusion-type relationship between $R^{\subset}$ and $(R^{\bullet})^{\subset}$, $(R^{\bullet})^{+}$, $(R^{+})^{\bullet}$, and considering $R^{\subset}$ alone as a partial order approximation of R is a little bit controversial (see an example from Figure 1 of [6]), but justifiable in some cases (see [6]).

Partial order approximations of arbitrary binary relations play a crucial role in the theory of non-numerical rankings based on the pairwise comparisons paradigm [4, 7].

Let $\mathcal{P}$ be the following set of properties *over the relation* R, $\mathcal{P} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$, where:

- $\alpha_1 \stackrel{df}{=} \forall a,b \in X.\ bR^{ref} \subset aR^{ref} \wedge R^{ref}a \subset R^{ref}b$, that is, $\alpha_1 =$ *inclusion property*,
- $\alpha_2 \stackrel{df}{=} \forall a,b,c \in X.\ aRb \wedge bRa \Longrightarrow aRc$, that is, $\alpha_2 =$ *transitivity*,
- $\alpha_3 \stackrel{df}{=} \forall a,b \in X.\ \neg(aR^{cyc}b)$, that is $\alpha_3 =$ *acyclicity*,
- $\alpha_4 \stackrel{df}{=} \alpha_1 \wedge \alpha_3$,
- $\alpha_5 \stackrel{df}{=} (\forall a \in X.\ \neg(aRa)) \wedge \alpha_2$, that is, $\alpha_5 =$ *partial ordering*.

Consider the property-driven rough set approximation space

$$(X \times X, \{P_\alpha \mid \alpha \in \mathcal{P}\}).$$

Directly from the definitions we may conclude that an arbitrary relation R has (see [6], Sections 3 and 4, for details):

- α_1-lower bound, but may not have α_1-upper bound,
- α_2-lower bound and α_2-upper bound,
- α_3-lower bound, but may not have α_3-upper bound,
- α_4-lower bound, but may not have α_4-upper bound,
- α_5-lower bound, but may not have α_5-upper bound,

We have here $\mathcal{P}^{\cap} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$, $\mathcal{P}^{\cup} = \emptyset$.

It turns out that all the partially ordered approximations $R^{\subset}$, $R^{\subset\wedge\bullet}$, $(R^{\bullet})^{\subset}$, $(R^{\bullet})^{+}$ and $(R^{+})^{\bullet}$ from Theorem 12.1 can be obtained naturally using the Rough Sets approach proposed in the previous sections. First, we will show that all operations involved are either lower or upper α-approximations.

Proposition 12.13 ([6])

1. $R^{\subset} = \underline{\mathbf{A}}_{\alpha_1}(R)$,
2. $R^{+} = \overline{\mathbf{A}}_{\alpha_2}(R)$,
3. $R^{\bullet} = \underline{\mathbf{A}}_{\alpha_3}(R)$,
4. $R^{\subset\wedge\bullet} = \underline{\mathbf{A}}_{\alpha_4}(R) = \underline{\mathbf{A}}_{\alpha_1\wedge\alpha_3}(R)$,

Proof (1) Since R has α_1-lower bound and $lb_{\alpha_1}(R) = \{R^{\subset}\}$, for the details of the latter see [6], Sections 3 and 4.
(2) Since $\alpha_2 \in \mathcal{P}^{\cap}$ and R has α_2-upper bound we can use Proposition 12.6(2), which says that $\overline{\mathbf{A}}_{\alpha_2}(R)$ is the smallest transitive relation containing R, that is, R^{+} (c.f. [12]).
(3) First note that R has α_3-lower bound. By the definition, we have

$\underline{\mathbf{A}}_{\alpha_3}(R) = \bigcap\{Q \mid Q \in max(lb_{\alpha_3}(R))\}$. Let $aR^{cyc}b$. This means $a = a_1Ra_2R\ldots Ra_{k-1}Ra_k = b$, where $i \neq j \Rightarrow a_i \neq a_j$. Let $Q \in lb_{\alpha_1}(R)$. Note that, $Q \in max(lb_{\alpha_1}(R))$ if and only if there is a_r such that $(a_{r-1}, a_r) \notin Q$ but $a = a_0Qa_1\ldots Qa_{r-1}$ and $a_rQ\ldots Qa_k$. Hence $R^{\bullet} = \bigcap\{Q \mid Q \in max(lb_{\alpha_3}(R))\} = \underline{\mathbf{A}}_{\alpha_3}(R)$.
(4) Since R has α_4-lower bound, $\underline{\mathbf{A}}_{\alpha_4}(R)$ exists and can be constructed. We have $\underline{\mathbf{A}}_{\alpha_4}(R) = \bigcap\{Q \mid Q \in max(lb_{\alpha_1\wedge\alpha_3}(R))\} = \bigcap\{Q \mid Q \in max(\{R^{\subset} \cap S \mid S \in lb_{\alpha_3}(R))\})\} = R^{\subset} \cap \bigcap\{Q \mid Q \in max(lb_{\alpha_3}(R))\}$. From (3) it follows $\bigcap\{Q \mid Q \in max(lb_{\alpha_3}(R))\} = R^{\bullet}$, so $R^{\subset\wedge\bullet} = R^{\subset} \cap R^{\bullet} = \underline{\mathbf{A}}_{\alpha_1\wedge\alpha_3}(R) = \underline{\mathbf{A}}_{\alpha_4}(R)$. □

Now we will show that compositions of appropriate approximations are allowed.

Proposition 12.14

1. α_1 *l-preserves* α_3.
2. α_2 *u-preserves* α_3.
3. α_3 *l-preserves* α_1 *and* α_3 *l-preserves* α_2.

Proof (1) By Theorem 12.1(1), if R is acyclic, $R^{\subset}$ is acyclic as well, and clearly R has α_1-lower bound ($\emptyset$ for example).
(2) By Theorem 12.1(1), transitivity preserves acyclity and of course R has α_2-lower bound ($\emptyset$ for example).
(3) By Theorem 12.1(1), acyclic refinement preserves both transitivity and inclusion property kernel, and of course R has α_2-lower bound (for example $\emptyset$ and $\mathcal{X}$).
□

The main result of this section can now be formulated as follows.

Proposition 12.15 ([6])

1. $(R^{\bullet})^{\subset} = \underline{\mathbf{A}}_{\alpha_1}(\underline{\mathbf{A}}_{\alpha_3}(R))$,
2. $(R^{\subset})^{\bullet} = \underline{\mathbf{A}}_{\alpha_3}(\underline{\mathbf{A}}_{\alpha_1}(R)) = R^{\subset}$, *so* $\underline{\mathbf{A}}_{\alpha_3}(\underline{\mathbf{A}}_{\alpha_1}(R)) = \underline{\mathbf{A}}_{\alpha_1}(R)$,
3. $(R^{\bullet})^{+} = \overline{\mathbf{A}}_{\alpha_2}(\underline{\mathbf{A}}_{\alpha_3}(R))$,
4. $(R^{+})^{\bullet} = \underline{\mathbf{A}}_{\alpha_3}(\overline{\mathbf{A}}_{\alpha_2}(R))$.

Proof From Propositions 12.13 and 12.14, and, for (2), from Theorem 12.1(3). □

Proposition 12.15 illustrates well the basic properties of property-driven rough set approximations of binary relations by partial orders. Below we provide some observations.

- Property α_5 does not appear in Proposition 12.15 at all. It is actually a rather useless property. No upper bound exists in a general case, and the relation $\underline{\mathbf{A}}_{\alpha_5}(R)$ is usually not very interesting (c.f. [6]). Property α_5 (being a partial order) is just too strong to be efficiently handled as a whole. We can get much better results when we treat the components of α_5, for instance acyclicity and transitivity, separately and then compose the results obtained.

- We have $R^{\subset\wedge\bullet} = \underline{\mathbf{A}}_{\alpha_4}(R) = \underline{\mathbf{A}}_{\alpha_1\wedge\alpha_3}(R) \subseteq (R^{\bullet})^{\subset} = \underline{\mathbf{A}}_{\alpha_1}(\underline{\mathbf{A}}_{\alpha_3}(R))$, and $R^{\subset\wedge\bullet} = \underline{\mathbf{A}}_{\alpha_1\wedge\alpha_3}(R) \subseteq \underline{\mathbf{A}}_{\alpha_3}(\underline{\mathbf{A}}_{\alpha_1}(R)) = (R^{\subset})^{\bullet} = R^{\subset}$ which illustrates Proposition 12.11(1).
- In general, $\underline{\mathbf{A}}_{\alpha_1}(\underline{\mathbf{A}}_{\alpha_3}(R)) = (R^{\bullet})^{\subset}$ and $\underline{\mathbf{A}}_{\alpha_3}(\underline{\mathbf{A}}_{\alpha_1}(R)) = (R^{\subset})^{\bullet} = R^{\subset}$ are *not equal*.
- We also have $(R^{\bullet})^{+} = \overline{\mathbf{A}}_{\alpha_2}(\underline{\mathbf{A}}_{\alpha_3}(R)) \subseteq \underline{\mathbf{A}}_{\alpha_3}(\overline{\mathbf{A}}_{\alpha_2}(R)) = (R^{+})^{\bullet}$. which illustrates Proposition 12.10(2).

While the approximations of arbitrary relations by partial orders, motivated by pairwise comparisons non-numerical ranking [4, 7], were initially defined in terms of the standard theory of relations, their Rough Sets versions better explain those definitions. The Rough Sets versions provide formal motivation and explanation in places where in the classical versions were just 'gut feelings'.

12.9 Approximations by Equivalence Relations

In this section we will apply the theory of property-driven rough approximations to the approximation of arbitrary binary relations by equivalence relations.

We will start with the classical well-known result.

Proposition 12.16 (Folklore, c.f. [2, 8, 12])
For every relation $R \subseteq X \times X$, the relations $((R^{ref})^{\overline{sym}})^{+} = ((R^{\overline{sym}})^{ref})^{+} = ((R^{\overline{sym}})^{+})^{ref}$ are equivalence relations, and $R \subseteq ((R^{ref})^{\overline{sym}})^{+}$. □

However, in general, the relation $((R^{ref})^{+})^{\overline{sym}} = ((R^{+})^{ref})^{\overline{sym}} = ((R^{+})^{\overline{sym}})^{ref}$ may not be an equivalence relation. The simplest example is probably $X = \{a,b,c\}$ and $R = \{(a,c),(b,c)\}$. Then, $((R^{ref})^{+})^{\overline{sym}} = \{(a,c),(b,c),(c,a),(c,b),(a,a),(b,b),(c,c)\}$, and this relation is not transitive as we have bRc and cRa, but not bRa.

Symmetric closure is not the only method for enforcing symmetry. We can also used the idea of 'kernel' for this task.

Definition 12.14. Let $R \subseteq X \times X$, we define

$$R^{\overline{sym}} = \{(a,b) \mid (a,b) \in R \wedge (b,a) \in R\}.$$

The relation $R^{\overline{sym}}$ will be called a **symmetric kernel** of R. □

It turns out that the symmetric kernel preserves transitivity and can also be used as a tool to construct an approximation by an equivalence relation.

Proposition 12.17. *For every $R \subseteq X \times X$, we have:*

1. *$R^{\overline{sym}}$ is symmetric and $R^{\overline{sym}} \subseteq R$.*
2. *$R^{\overline{sym}} = \bigcup\{Q \mid Q \subseteq R$ and Q is symmetric $\}$.*

3. $((R^{ref})^{\overline{sym}})^{+} = ((R^{\overline{sym}})^{ref})^{+} = ((R^{\overline{sym}})^{+})^{ref}$.
4. $((R^{ref})^{+})^{\overline{sym}} = ((R^{+})^{ref})^{\overline{sym}} = ((R^{+})^{\overline{sym}})^{ref}$.
5. $((R^{ref})^{\overline{sym}})^{+}$ *is an equivalence relation.*
6. $((R^{ref})^{+})^{\overline{sym}}$ *is an equivalence relation.*

Proof. (1) From the definition.
(2) It suffices to show that if $Q \subseteq R$ and Q is symmetric, then $Q \subseteq R^{\overline{sym}}$. Let $(a,b) \in Q$, so $(a,b) \in R$. If $(b,a) \notin R$ then $(b,a) \notin Q$, and, since Q is symmetric, $(a,b) \notin Q$. Hence $(b,a) \in Q$. Since $Q \subseteq R$, $(b,a) \in R$. But if $(a,b) \in R$ and $(b,a) \in R$, then $(a,b) \in R^{\overline{sym}}$. Hence $Q \subseteq R^{\overline{sym}}$.
(3) and (4). Reflexive closure is just adding the identity relation. It does not interfer with either transitive closure or symmetric kernel operation.
(5) Since transitive closure preserves symmetry (c.f. [12]).
(6) It suffices to show that symmetric kernel preserves transitivity. Define $Q = (R^{ref})^{+}$. Clearly Q is transitive. Suppose that $Q^{\overline{sym}}$ is not. This means thare are $a,b,c \in X$ such that $aQ^{\overline{sym}}b$ and $bQ^{\overline{sym}}c$ but $\neg(aQ^{\overline{sym}}c)$. By Definition 12.14, $Q^{\overline{sym}} \subseteq Q$, so aQb and bQc. The relation Q is transitive, so we also have aQc. Since $\neg(aQ^{\overline{sym}}c)$, then, from Definition 12.14, $\neg(cQ^{\overline{sym}}a)$. However, as $Q^{\overline{sym}}$ is symmetric, $aQ^{\overline{sym}}b$ and $bQ^{\overline{sym}}c$ means that we also have $bQ^{\overline{sym}}a$ and $cQ^{\overline{sym}}b$, and consequently bQa and cQb. Since Q is transitive, then have cQa. But aQc and cQa implies that $aQ^{\overline{sym}}c$ and $cQ^{\overline{sym}}a$, contradicting $\neg(aQ^{\overline{sym}}c)$. Hence $Q^{\overline{sym}}$ is transitive. □

The relations $(R^{\overline{sym}})^{+}$, $(R^{+})^{\overline{sym}}$, $(R^{\overline{sym}})^{+}$ and Proposition 12.17 are illustrated by an example in Figure 12.2. While $R \subseteq (R^{\overline{sym}})^{+}$, there is no universal inclusion relationship between R and neither $(R^{\overline{sym}})^{+}$ nor $(R^{+})^{\overline{sym}}$.
Let $\mathcal{P}$ be the following set of properties *over the relation* R, $\mathcal{P} = \{\beta_1, \beta_2, \beta_3, \beta_4\}$, where:

- $\beta_1 \stackrel{df}{=} \forall a \in X.\ aRa$, that is, $\beta_1 =$ *reflexivity*,
- $\beta_2 \stackrel{df}{=} \forall a,b \in X.\ aRb \Rightarrow bRa$, that is, $\beta_2 =$ *symmetry*,
- $\beta_3 \stackrel{df}{=} \forall a,b,c \in X.\ aRb \wedge bRa \Longrightarrow aRc$, that is, $\beta_3 =$ *transitivity*,
- $\beta_4 \stackrel{df}{=} \beta_1 \wedge \beta_2 \wedge \beta_3$, that is, $\beta_3 =$ *equivalence relation*.

Consider the property-driven rough set approximation space

$$(X \times X, \{P_\alpha \mid \alpha \in \mathcal{P}\}).$$

Directly from the definitions, we may conclude that an arbitrary relation $R \subseteq X \times X$ has:

- β_1-upper bound, but may not have β_1-lower bound, β_1-lower bound exists only when R is already reflexive,
- β_2-upper bound, but may not have β_2-lower bound, β_2-lower bound exists only when $R \cap Id \neq \emptyset$,
- β_3-lower bound and β_3-upper bound,

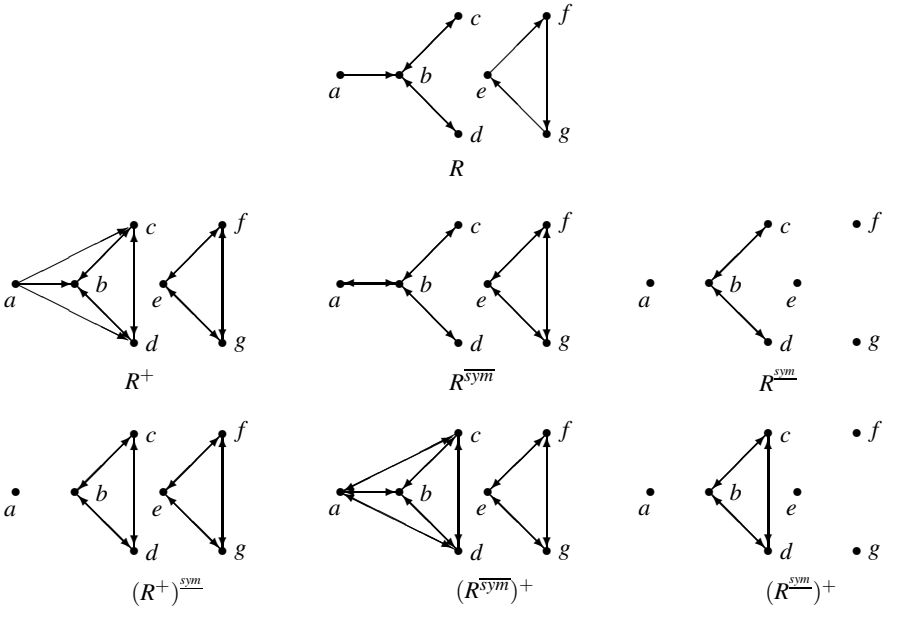


Fig. 12.2. An example of a relation R and the results of applications of transitive closure, symmetric closure and symmetric kernel, in various orders. If R is reflexive then, the relations $(R^+)^{\overline{sym}}$, $(R^{\overline{sym}})^+$ and $(R^{\overline{sym}})^+$ are equivalence relations. In this figure $a \bullet\!\!\longleftrightarrow\!\!\bullet b$ means $a \bullet\!\!\longrightarrow\!\!\bullet b$ and $a \bullet\!\!\longleftarrow\!\!\bullet b$.

- β_4-upper bound, but may not have β_4-lower bound, β_4-lower bound exists only when R is reflexive,

We have here $\mathcal{P}^{\cap} = \{\beta_1, \beta_2, \beta_3, \beta_4\}$ and $\mathcal{P}^{\cup} = \{\beta_1, \beta_2\}$.

It turns out that all the approximations by equivalence relations from Propositions 12.16 and 12.17 can naturally be obtained using the Rough Sets approach proposed in the previous sections. First, we will show that all operations involved are either lower or upper α-approximations.

Proposition 12.18. *Let R be a relation on X*

1. $R^{ref} = \overline{\mathbf{A}}_{\beta_1}(R)$.
2. $R^{\overline{sym}} = \overline{\mathbf{A}}_{\beta_2}(R)$.
3. $R^{\underline{sym}} = \underline{\mathbf{A}}_{\beta_2}(R)$.
4. $R^{+} = \overline{\mathbf{A}}_{\beta_3}(R)$.

Proof. (1) Clearly R has β_1-upper bound. We have $ub_{\beta_1}(R) = \{Q \mid R^{ref} \subseteq Q\}$, Hence, $min(ub_{\beta_1}(R)) = \{R^{ref}\}$.

(2) Since $\beta_2 \in \mathcal{P}^{\cap}$ and R has β_2-upper bound we can use Proposition 12.6(2), which says that $\overline{\mathbf{A}}_{\beta_2}(R)$ is the smallest symmetric relation containing R, that is, $R^{\overline{sym}}$ (c.f. [12]).
(3) Since $\beta_2 \in \mathcal{P}^{\cup}$ and R, has β_2-lower bound we can use Proposition 12.6(1), which says that $\overline{\mathbf{A}}_{\beta_2}(R)$ is the greatest symmetric relation included R. From Proposition 12.17(2), it is $R^{\underline{sym}}$.
(4) Since $\beta_3 \in \mathcal{P}^{\cap}$ and R has β_3-upper bound we can use Proposition 12.6(2), which says that $\overline{\mathbf{A}}_{\beta_3}(R)$ is the smallest transitive relation containing R, that is, R^+ (c.f. [12]). □

We will now show that the compositions of appropriate approximations are allowed.

Proposition 12.19

1. β_2 *l-preserves and u-preserves* β_1.
2. β_3 *u-preserves* β_1.
3. β_2 *l-preserves* β_3 *but* β_2 *does* not *u-preserve* β_3.
4. β_3 *u-preserves* β_2.

Proof (1) Clearly if R is reflexive, then $R^{\underline{sym}}$ and $R^{\overline{sym}}$ are also reflexive. This and Proposition 12.18(2) and (2) prove this assertion.
(2) If R is reflexive, then R^+ is also reflexive. Hence, by Proposition 12.18(4) β_3 u-preserves β_1.
(3) If R is transitive, then by Proposition 12.17(6), $R^{\underline{sym}}$ is also transitive. Hence, by Proposition 12.18(3), β_2 l-preserves β_3. However, $R^{\overline{sym}}$ may not be transitive, as the example after Proposition 12.16 shows. Hence by Proposition 12.18(2), β_2 does not u-preserve β_3.
(4) If R is symmetric, then R^+ is symmetric too. Hence, by Proposition 12.18(4), β_3 u-preserves β_2. □

We can now present the main result of this section.

Proposition 12.20. *Let R be a relation on X.*

1. $((R^{ref})^{\overline{sym}})^+ = \overline{\mathbf{A}}_{\beta_3}(\overline{\mathbf{A}}_{\beta_2}(\overline{\mathbf{A}}_{\beta_1}(R)))$.
2. $((R^{ref})^{\underline{sym}})^+ = \overline{\mathbf{A}}_{\beta_3}(\underline{\mathbf{A}}_{\beta_2}(\overline{\mathbf{A}}_{\beta_1}(R)))$.
3. $((R^{ref})^+)^{\underline{sym}} = \underline{\mathbf{A}}_{\beta_2}(\overline{\mathbf{A}}_{\beta_3}(\overline{\mathbf{A}}_{\beta_1}(R)))$.
4. $((R^{ref})^{\underline{sym}})^+ \subseteq ((R^{ref})^+)^{\underline{sym}}$.

Proof. (1), (2) and (3) A simple consequence of Propositions 12.18 and 12.19.
(4) From Proposition 12.10(2) we have
$\overline{\mathbf{A}}_{\beta_3}(\underline{\mathbf{A}}_{\beta_2}(\overline{\mathbf{A}}_{\beta_1}(R))) \subseteq \underline{\mathbf{A}}_{\beta_2}(\overline{\mathbf{A}}_{\beta_3}(\overline{\mathbf{A}}_{\beta_1}(R)))$. □

Proposition 12.20 illustrates well the basic properties of property-driven rough set approximations of binary relations by equivalence relations. Below, we provide some observations.

- Property β_4 does not appear in Proposition 12.20 at all. It is actually a rather useless property. Quite often $\underline{\mathbf{A}}_{\beta_4}(R) = \emptyset$ and $\overline{\mathbf{A}}_{\beta_4}(R) = \mathbf{X}$, which is not very helpful. The property β_4 (being an equivalence relation) is just too strong to be efficiently handled as a whole. We can get much better results when we treat the components of β_4, reflexivity, symmetry and transitivity, separately and then compose them together in appropriate manner.
- The assertion $((R^{ref})^{\overline{sym}})^+ \subseteq ((R^{ref})^+)^{\overline{sym}}$ can of course be proved independently, without using Rough sets, but Proposition 12.10(2) makes this (otherwise not obvious) proof trivial.
- We have applied the reflexive closure first, but in fact, it can be applied as the second or third as well (see Propositions 12.16, 12.17(3) and 12.17(4)).

Standardly, $((R^{ref})^{\overline{sym}})^+$ is considered as the only approximation of R by an equivalence relation [12]. In the Rough sets approach, it is natural to think of both upper and lower approximations, which in this case leads to the design of $((R^{ref})^{\overline{sym}})^+$ and $((R^{ref})^+)^{\overline{sym}}$ approximations.

12.10 Final Comment

The approach presented in this chapter is called *property-driven* as its main purpose is to find an approximation, either lower or upper, that satisfies a given predicate, called a property. It could be seen as a substantial extension of the ideas presented for relations in [16, 17] and specially recently in [6]. Both this chapter and [6] were motivated by problems occurring when non-numerical ranking is constructed from empirical data [4, 7]. When thinking in terms of *properties*, very often either only lower or only upper approximation does make sense, and quite often *neither of them if the property is too sophisticated*. This lead us to the idea of composite and mixed approximations.

Proposition 12.11 might be the most useful result of this chapter as it indicates how properties should be dealt with to get the best approximations.

We would like to point out that all the assumptions from Section 12.4, especially Assumption 1, relate only to elementary properties. The requirements for composite properties are indirect and so much weaker.

We believe the schedules can often be interpreted as a simulation of 'property-driven non-numerical metrics', and that finding a good schedule means finding a good approximation. But finding a good schedule appears to be more art than science, as our experience with partial orders and equivalence relations indicates.

In general, for a proper schedule s, we usually have $R \setminus \mathbf{A}^s(R) \neq \emptyset$, $\mathbf{A}^s(R) \setminus R \neq \emptyset$ and $R \cap \mathbf{A}^s(R) \neq \emptyset$. The formal definition of the 'best' proper schedule is an open problem. However, we believe that any rule, if proposed, could only be treated as a guide, as the problem seems to be very domain related.

In this chapter, we deal only with single n-ary relations and the composite properties are of the form $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_k$. A natural extension of the presented theory would be allowing composite properties with the operators of conjunction and

negation as well. Another natural extension would be to allow properties with more than one relational symbol, that is, an extension to the relational systems (a tuple $(X, R_1, ..., R_k)$, where X is a set and $R_1, ..., R_k$ are relations on X, c.f. [2]), as suggested in [5]. For the former extension, we see some technical problems. For the latter extension, while the extension of general theory is not problematic, one just needs to follow [5], we expect plenty of technical problems for particular applications. While in theory any abstract data type (as defined for example in [1]), can be represented by a relational structure, it is seldom done in practice, as much of the intution is then lost. From the applications point of view, an extension of the ideas presented here to (at least some of) abstract algebras [2], would be very helpful.

Acknowledgements. The author is grateful to the anonymous referees, who pointed out many typos and suggested a better way to organise the materials of this chapter, and would like to thank Andrzej Skowron and Zbigniew Suraj for their support and encouragement during the work on this chapter.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: Data Structures and Algorithms. Addison-Wesley (1983)
2. Burris, S., Sankappanavar, H.P.: A Course in Universal Algebra. Springer (1981)
3. Huth, M.R.A., Ryan, M.D.: Logic in Computer Science. Cambridge University Press (2000)
4. Janicki, R.: Pairwise comparisons based non-numerical ranking. Fundamenta Informaticae 94(2), 197–217 (2009)
5. Janicki, R.: On Rough Sets with Structures and Properties. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS(LNAI), vol. 5908, pp. 109–116. Springer, Heidelberg (2009)
6. Janicki, R.: Approximations of arbitrary binary relations by partial orders. Classical and rough set models. In: Peters et al. [11], pp. 17–38
7. Janicki, R., Zhai, Y.: On a pairwise comparison based consistent non-numerical ranking. Logic Journal of IGPL (2011), doi:10.1093/jigpal/jzr018
8. Kuratowski, K., Mostowski, A.: Set Theory, 2nd edn. North-Holland (1976)
9. Pawlak, Z.: Rought Sets. International Journal of Computer and Information Sciences 34, 557–590 (1982)
10. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. System Theory, Knowledge Engineering and Problem Solving, vol. 9, Kluwer Academic Publishers, Dordrecht (1991)
11. Peters, J.F., Skowron, A., Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.): Transactions on Rough Sets XIII. LNCS, vol. 6499. Springer, Heidelberg (2011)
12. Rosen, K.H.: Discrete Mathematics and Its Applications. McGraw-Hill, New York (1999)
13. Skowron, A., Stepaniuk, J.: Approximation of relations. In: Ziarko, W. (ed.) Rough Sets, Fuzzy Sets and Knowledge Discovery, Workshops in Computing, pp. 161–166. Springer & British Computer Society, London, Berlin (1994)

14. Skowron, A., Stepaniuk, J.: Tolarence approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
15. Stewart, J.: Calculus. Concepts and Contexts. Brooks/Cole (1997)
16. Yao, Y.Y.: Two views of the theory of rough sets in finite universes. International Journal of Approximate Reasoning 15, 291–317 (1996)
17. Yao, Y.Y., Wang, T.: On Rough Relations: An Alternative Formulation. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 82–91. Springer, Heidelberg (1999)

Chapter 13
Towards a Comprehensive Similarity Analysis of Voting Procedures Using Rough Sets and Similarity Measures

Janusz Kacprzyk, Hannu Nurmi, and Sławomir Zadrożny

To Professor Zdzisław Pawlak, a friend and mentor, who has invented rough sets, a tool combining an ingenious simplicity with an extraordinary conceptual strength, intuitive appeal, and application potential

Abstract. An interesting and important problem of how similar (and/or dissimilar) are the voting procedures (social choice functions) is dealt with. First, we extend our previous qualitative-type analysis based on rough sets theory which makes it possible to partition the set of voting procedures considered into some subsets within which the voting procedures are indistinguishable, that is, (very) similar. Then, we propose an extension of those analyses towards a quantitative evaluation of a degree of similarity. We use some known measures of similarity and dissimilarity for binary

Janusz Kacprzyk
Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl
and
Warsaw School of Information Technology (WIT)
ul. Newelska 6, 01-447 Warsaw, Poland

Hannu Nurmi
Department of Political Science and Contemporary History, University of Turku
FI-20014 Turku, Finland
e-mail: hnurmi@utu.fi

Sławomir Zadrożny
Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
e-mail: zadrozny@ibspan.waw.pl
and
Technical University of Radom
ul. Malczewskiego 29, 26-600 Radom, Poland

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 359–380.
springerlink.com

patterns (strings) exemplified by the one based on the Hamming distance and that due to Jaccard-Needham. The use of both the measures makes it possible to obtain a quantitative evaluation of how similar the particular voting procedures are. This quantitative evaluation, when combined with a more qualitative evaluation obtained by using rough sets, provide a comprehensive view of similarity.

Keywords: Social choice function, voting procedure.

13.1 Introduction

This chapter deals with the problem of *voting procedures* that are in this chapter considered more in the context of political elections, that is, social systems, than multi-agent systems in which voting has been widely employed and is constantly gaining importance – cf. for a comprehensive exposure in particular Pitt et al. [42] but also Pitt et al. [40, 41].

The voting procedures are perhaps the best known and most intuitively appealing examples of *social choice functions*. A voting procedure is meant to determine the winner of an election as a function of votes cast by the voters. Obviously, the votes express the individual voter preferences which usually differ. Social choice theorists have developed a huge number of various procedures (social choice functions) and criteria for their evaluation (cf. Arrow, Sen and Suzumura [2], Kelly [20], Plott [43], Schwartz [48], etc.)

A general setting is as follows: we have n, $n \geq 2$ individuals who present their testimonies over the set of m, $m \geq 2$, options, assumed here to be, for instance, *individual preference relations* defined as binary preferences over the set of options, orderings over the set of options, etc., and we focus on *social choice functions*, a class of social choice procedures that select a set of winning alternatives, that is, a set of options that best reflects the opinions of the individuals, as a function of individual preferences (or orderings) over the set of options. This may clearly be easily recast in terms of voting.

A major concern in social choice (voting) theory has been the problem whether and to which extent the social choice (voting) procedures do or do not satisfy some plausible and reasonable axioms and conditions. Perhaps the best known example is here the famous Arrows theorem, and a multitude paradoxes of voting. We will not deal here with this problem, and refer the reader for more information to, for instance, Arrow [1], Gibbard [12], Kelly [19], May [22], Nurmi [27], Riker [46], Satterthwaite [47], etc.

An important problem is also how similar or dissimilar (depending on the context and/or purpose) the particular voting procedures are. Except for a foundational reference book by Nurmi [27], the papers relevant in this respect, and which basically follow to some extent our line of reasoning, may be exemplified by: Baigent [3], Elkind, Faliszewski and Slinko [5], McCabe-Dansted and Slinko [23], Richelson [45], etc.

The second purpose of this chapter is to show that for the analysis of how similar (or dissimilar) the particular voting procedures are Pawlak's *rough sets* (cf. Pawlak [33, 34] or Pawlak and Skowron [36]) may provide a set of very useful tools and techniques which are, on the one hand, simple and intuitively appealing, and – on the other hand – give much insight. Basically, we consider the voting procedures and look at some set of popular and well-established criteria against which the voting procedures are evaluated. However, by using some rough sets based analysis we will first find a subset of criteria that are most characteristic for the particular procedure, that is, those which really differentiate the particular voting procedures, and then proceed with a similarity type analysis. The idea was proposed first by Fedrizzi, Kacprzyk and Nurmi [10], and we will extend it here to obtain more insight that might be interesting for the readers interested or working in social choice and voting theory. This purpose has implied that we have employed a limited set of rough sets tools and techniques to make the chapter comprehensible to an audience with possibly limited familiarity with rough sets. One should mention that Pawlak [35] himself was interested in conflict analyses and resolution, and proposed an original rough sets based framework. Although it is different than the one used here for the analysis of similarity of the voting procedures, some elements of it are to some extent close in spirit with the philosophy of our approach.

In our analysis we will follow what might be termed an indiscernibility based analysis so that in our evaluation of the similarity of voting procedures we will use first of all "qualitative" rough sets analysis. However, it will give us much insight. That analysis will then be augmented with the use of some basic quantitative measures of similarity of binary patterns, namely those based on the Hamming distance and the Jaccard-Needham mesure. This will give us a quantitative view of the similarity of the voting procedures. However, a comprehensive view of that similarity is first obtained by the use of the combination of both the approaches employed.

The use of other, more sophisticated measures of similarity (for instance, those due to Dice, Yule, Russell-Rao, Sokal-Michener, Rogers-Tanimoto, and Kulczyński – cf. Tubbs [50]), both of those which satisfy the metric condition and not, is beyond the scope of this chapter and will be the subject of next papers.

Notice that this approach is different from the approach by Kacprzyk and Zadrożny [17], [18] in which some distinct classes of voting procedures are determined using the concept of an ordered weighted averaging (OWA) aggregation operator (cf. Yager and Kacprzyk [51], Yager, Kacprzyk and Beliakov [52]) in which the change of the order of variables to be aggregated and the type of weights (i.e. the aggregation behavior) determines various classes of voting procedures.

To summarize, this chapter is a tribute to Professor Zdzisław Pawlak, who has invented rough sets theory that has combined an ingenious simplicity with an extraordinary conceptual power that can help solve problems in so many different and seemingly distant areas, in voting and social choice in our case.

13.2 A Brief Introduction to the Theory of Rough Sets

In this section we will briefly recall some basic concepts and properties of rough sets theory which may be useful for the social choice and voting community to follow the discussion in this chapter. For more information on rough sets, we refer the reader to, for instance, the seminal paper by Pawlak [33], and other sources exemplified by Pawlak himself [34], Polkowski (*e.g.*, [44]), Skowron (*e.g.*, [36], Pawlak and Skowron, [37], [38], [39],), and Słowiński (*e.g.*, [13]) and their collaborators).

Basically, the concept of a *rough set* provides a conceptually simple and efficient tool for the representation and processing of imprecise knowledge. Roughly speaking, the concept of a rough set is employed in the contexts where the classes into which the objects are to be classified are imprecise but can nevertheless be approximated by precise sets, from above and below.

Suppose that $U = \{u\}$ is a *universe of discourse*. It can usually be partitioned in various manners into a family $\mathbf{R}$ of partitionings or equivalence relations defined on U. A *knowledge base*, denoted by K, is the pair $K = (U, \mathbf{R})$. Let now $\mathbf{P}$ be a non-empty subset of $\mathbf{R}$, $\mathbf{P} \subset \mathbf{R}, \mathbf{P} \neq \emptyset$. Then, the intersection of all equivalence relations (or, in other words, partitionings) in $\mathbf{P}$, which is also an equivalence relation, is called an *indiscernibility relation* over $\mathbf{P}$, denoted by $IND(\mathbf{P})$ and the family of its equivalence classes is termed the $\mathbf{P}$-basic knowledge about U in knowledge base K. Clearly, it represents all that can be said about the elements of the universe of discourse U under $\mathbf{P}$. That is, one cannot classify the elements of U any deeper than to the equivalence classes of $IND(\mathbf{P})$. For instance, if U consists of some objects and $\mathbf{P} = \{R_1, R_2\}$ such that R_1 partitions the objects into the classes labeled "heavy" and "lightweight", and R_2 partitions into the classes labeled "black" and "white", then all that can be said about any element of U is that it belongs to some of the four combinations of particular classes: "heavy-and-black", "heavy-and-white","lightweight-and-black", "lightweight-and-white" that are implied by U as shown above.

Equivalence classes of $IND(\mathbf{P})$ are called the *basic categories (concepts) of knowledge P*. If $Q \in \mathbf{R}$, that is, Q is an equivalence relation on U, then its equivalence classes are called *Q-elementary categories (concepts) of knowledge* $\mathbf{R}$.

Now, suppose that $X \subset U$, and R is an equivalence relation on U. Then, X is called *R-definable* or *R-exact* if it is a union of some R-elementary categories (R-basic categories); otherwise, it is called R-rough.

Rough sets can be approximately defined by associating with any $X \subset U$ and any equivalence relation R on U the following two sets (U/R denotes the set of all equivalence relations of R):

- a so-called *lower approximation* of X:

$$R_L X = \bigcup \{Y \in U/R \mid Y \subset X\} \tag{13.1}$$

- a so-called *upper approximation* of X:

$$R_U X = \bigcup \{Y \in U/R \mid Y \cap X \neq \emptyset\} \tag{13.2}$$

and a *rough set* is defined as the pair (R_L, R_U).

The meaning is obvious as the lower approximation yields those classes of equivalence relation R which are subsets of X, that is, it contains those elements which are necessarily also elements of X, while the upper approximation yields those classes of equivalence relation R which have at least one common element with X.

For our purpose the following two concepts related to the reduction of knowledge, that is, their effective and efficient expression, are crucial. First, suppose that we have a family of equivalence relations $\mathbf{R}$ on U, and one of its elements Z. Then, Z is called *dispensable* in $\mathbf{R}$ if

$$IND(\mathbf{R}) = IND(\mathbf{R} \setminus \{Z\}) \tag{13.3}$$

and otherwise it is called indispensable. If each Z in $\mathbf{R}$ is indispensable, then $\mathbf{R}$ is called *independent*.

Suppose now that we have a family of equivalence relations, $\mathbf{R}$, and its subfamily, $\mathbf{Q} \subset \mathbf{R}$. If

- $\mathbf{Q}$ is independent, and
- $IND(\mathbf{Q}) = IND(\mathbf{R})$,

then $\mathbf{Q}$ is called a *reduct* of $\mathbf{R}$; it is clear that the reducts need not be unique.

The *core* of $\mathbf{R}$ is the set of all indispensable equivalence relations in $\mathbf{R}$. Obviously, the core of $\mathbf{R}$ is the intersection of all reducts of $\mathbf{R}$ – cf. Pawlak [33].

With respect to knowledge reduction, the concepts of a core and reduct can be viewed as follows. The core consists of those classifications (equivalence relations) which are most essential in the knowledge available, that is, no equivalence relation that belongs to the core can be discarded in the knowledge reduction process (aimed at a more effective and efficient representation) without distorting the knowledge itself. A reduct, on the other hand, yields a set of equivalence relations which is sufficient for the characterization of knowledge available without losing anything relevant.

An interesting method for the determination of the reducts and cores using Boolean reasoning was proposed by Pawlak [34], and Pawlak and Skowron [38].

Notice that in this chapter we will deal with analyses based on indiscernibility relations. In rough sets theory, the concept of a discernibility relation (cf. Yao and Zhao [53] is also employed and can be used in the context of similarity analysis of the voting procedures but this interesting approach will not be considered in this papers.

These basic elements of rough sets theory will now be employed in the problem considered.

13.3 A Comparison of Voting Procedures

There is a rich literature on social choice functions, notably voting procedures, and – as we have already indicated in Section 13.1 – there is a multitude of various

social choice functions (voting procedures), both simple and sophisticated, old and new, intuitively appealing and not, widely employed and not, etc. There is no clear "winner" as one can easily show examples when any particular procedure does not yield good results. The problem of comparison and evaluation of voting procedures (social choice functions) has been widely studied in the literature and the interested reader is referred to, for instance, Richelson [45], Straffin [49], Nurmi [27].

A simple and intuitive approach to the comparison of voting procedures (social choice functions), using rough sets as a point of departure, has been proposed by Fedrizzi, Kacprzyk and Nurmi [10]. This chapter is based on that idea, and extends it.

First, to keep our discussion clear and constructive and to better present the idea of our approach, we will assume here the basic framework adopted by Fedrizzi, Kacprzyk and Nurmi [10], and refer our analysis directly to an example of specific voting procedures and evaluation criteria considered. Their choice is well motivated by results obtained in social choice and voting theory.

Suppose that we take the following 13 popular and widely used voting procedures (social choice functions), and very briefly summarize their essence:

1. Amendment: proposals (options) under consideration are paired with the status quo; and if a variation on the proposal is introduced, then it is paired with the proposal and voted on as an amendment prior to the final vote, then, if the amendment succeeds, the amended proposal is eventually paired with the status quo in the final vote, otherwise, the amendment is eliminated prior to the final vote; this is a very specific voting procedure.
2. Copeland: selects the option with the largest so-called Copeland score which is the number of times an option beats other options minus the number of times that option loses to other options in pairwise comparisons,
3. Dodgson: each voter gives a rank ordered list of all options, from the best to worst, and the winner is the option for which we need to perform the minimum number of pairwise swaps (added over all candidates) before they become a Condorcet winner, that is, an option that defeats every other option in pairwise contests with a majority of votes.
4. Schwartz: selects the set of options over which the collective majority preferences are cyclic and the entire cycle is preferred over the other options; it is the single element in case there is a Condorcet winner, otherwise it consists of several options,
5. Max-min: selects the option for which the minimal support in all pairwise comparisons is the largest,
6. Plurality: each voter selects one option (or none in the case of abstention), and the options with the most votes win,
7. Borda: each voter provides a linear ordering of the options which are assigned a score (the so-called Borda score) as follows: if there are n candidates, $n-1$ points are given to the first ranked option, $n-2$ to the second ranked, etc.; these numbers are summed up for each option to end up with the Borda score for that option, and the option(s) with the highest Borda score win(s).

8. Approval: each voter selects a subset of the candidate options and the option(s) with the most votes is/are the winner(s).
9. Black: selects the Condorcet winner, that is, an option that beats or ties all others in pairwise comparisons, when one exists and the Borda count winner (as described above) in the absence of a Condorcet winner,
10. Runoff: the option ranked first by more than half of the electorate is chosen if one exists. Otherwise, the two options ranked first by more voters than any other option are compared with each other and the winner is the one ranked first (among the remaining options) by more voters than the other option.
11. Nanson: we iteratively use the Borda count, at each step dropping the candidate with the smallest score (majority)[1],
12. Hare: the ballots are linear orders over the set of options, and we repeatedly delete the options which receive the fewest number of first places in the votes, and the option(s) that remain(s) are declared as the winner(s),
13. Coombs: each voter rank orders all of the options, and if one option is ranked first by an absolute majority of the voters, then it is the winner, otherwise, the option which is ranked last by the largest number of voters is eliminated, and the procedure is repeated.

With respect to the criteria against which the voting procedures are to be compared, there is a huge number of them, but many voting theorists consider the following ones to be relevant (cf. Nurmi [27]):

1. Majority winner criterion: if there exists a majority (at least a half) that ranks a single option at the top (the first), higher than all other candidates, that option should win.
2. Monotonicity criterion: it is impossible to cause a winning option to lose by ranking it higher, or to cause a losing option to win by ranking it lower.
3. Consistency criterion: if the electorate is divided in two and an option wins in both parts, it should win in general.
4. Weak Pareto criterion: whenever all voters rank an option higher than another option, the latter option should never be chosen.
5. Participation criterion: to vote honestly should be better than not to vote at all.
6. Condorcet winner criterion: if an option beats every other option in pairwise comparisons, it should always win.
7. Condorcet loser criterion: if an option loses to every other option in pairwise comparisons, it should always loose.
8. Independence of irrelevant alternatives: if an option is added or removed, the relative rankings of the remaining options should remain the same.
9. Independence of clones: the outcome should be the same if we add options identical to the existing ones.
10. Reversal symmetry: if individual preferences of each voter are inverted, the original winner should never win.

[1] This may in fact be considered to be a modified version of the Nanson rule, cf. Fishburn [11].

11. Heritage criterion: if an option is chosen from the entire set of options using a particular voting procedure, then it should also be chosen from all subsets of the set of options (to which it belongs) using the same voting procedure and under the same preferences.
12. Polynomial time: it should be possible to find the winner in polynomial time with respect to the number of options and voters.

In our analyses we will only use the following 7 criteria, which are often consider the most relevant, denoted as follows:

1. A – Condorcet winner,
2. B – Condorcet loser,
3. C – majority winner,
4. D – monotonicity,
5. E – weak Pareto winner,
6. F – consistency, and
7. G – heritage.

We shall focus on the following 13 procedures:

1. Amendment,
2. Copeland,
3. Dodgson,
4. Schwartz,
5. Max-min,
6. Plurality,
7. Borda,
8. Approval,
9. Black,
10. Runoff,
11. Nanson,
12. Hare,
13. Coombs.

The point of departure is the "state of the art" presented in Table 13.1 which shows which voting procedures satisfies which criterion ("0" stands for "does not satisfy", and "1" stands for "satisfies").

It is easy to imagine that the data in Table 13.1 can be directly used for the comparison of the 13 voting procedures considered against the 7 criteria assumed by a pairwise comparison of rows. However, such a straightforward approach would not provide any deeper insight into the very differences between the voting procedures as the comparison would concern just particular voting procedures and not their more or less homogeneous classes which are clearly really interesting. This problem is closely related to Kacprzyk and Zadrożny's [17], [18] OWA operator based approach to the classification of voting procedures into a number of more general classes that are related, first, to the order in which the aggregation via an OWA operator proceeds and, second, to specific sets of weights of the respective OWA operators.

Table 13.1. Satisfaction of 7 criteria by 13 voting procedures

Voting procedure	Criteria						
	A	B	C	D	E	F	G
Amendment	1	1	1	1	0	0	0
Copeland	1	1	1	1	1	0	0
Dodgson	1	0	1	0	1	0	0
Schwartz	1	1	1	1	0	0	0
Max-min	1	0	1	1	1	0	0
Plurality	0	0	1	1	1	1	0
Borda	0	1	0	1	1	1	0
Approval	0	0	0	1	0	1	1
Black	1	1	1	1	1	0	0
Runoff	0	1	1	0	1	0	0
Nanson	1	1	1	0	1	0	0
Hare	0	1	1	0	1	0	0
Coombs	0	1	1	0	1	0	0

13.4 Equivalent Voting Procedures and Indispensable Criteria

Therefore, we will first try to obtain a more compact representation of the voting procedures and the satisfaction of criteria, operating for clarity on the example considered shown in Table 13.1.

First, let us concentrate on the crucial properties or attributes of the voting procedures by merging them into one class such that if they satisfy the same properties within the set of criteria, they may be considered to be equivalent, that is, belong to one class.

Thus, for example, the amendment and Schwartz' procedures have identical properties in Table 13.1. Hence, we can delete one of them. Similarly, Copeland's and Black's ones have identical properties in the table. The same holds for the runoff, Hare's and Coombs' ones. Therefore, this implies that by deleting all but one in each of the above set of identical rows (voting procedures), we obtain Table 13.2.

Hence, since each row is now different, we have classified the voting procedures considered into 9 "really different" (classes of) voting procedures:

1. Amendment (which stands now for Amendment and Schwartz),
2. Copeland (which stands now for Copeland and Black),
3. Dodgson,
4. Max-min,
5. Plurality,
6. Borda,
7. Approval,
8. Runoff (which stands now for Runoff, Hare and Coombs).
9. Nanson.

Table 13.2. Satisfaction of 7 criteria by 9 equivalent (classes of) voting procedures

Voting procedure	Criteria						
	A	B	C	D	E	F	G
Amendment	1	1	1	1	0	0	0
Copeland	1	1	1	1	1	0	0
Dodgson	1	0	1	0	1	0	0
Max-min	1	0	1	1	1	0	0
Plurality	0	0	1	1	1	1	0
Borda	0	1	0	1	1	1	0
Approval	0	0	0	1	0	1	1
Runoff	0	1	1	0	1	0	0
Nanson	1	1	1	0	1	0	0

Continuing in that spirit, we will try to further reduce the set of data characterizing the voting procedures and criteria by trying to find the *indispensable criteria*.

Just to recall this concept, if we have a family of equivalence relations $\mathbf{R}$ on U, and one of its elements Z, then Z is called *dispensable* in $\mathbf{R}$ if

$$IND(\mathbf{R}) = IND(\mathbf{R} \setminus \{Z\})$$

and otherwise it is called indispensable. If each Z in $\mathbf{R}$ is indispensable, then $\mathbf{R}$ is called *independent*. Notice that we consider here each attribute (which corresponds to a criterion) to be generating such an equivalence relation that to the same class there belong those voting procedures that fulfill this criterion, and to another class those which do not. This process is done by eliminating the criteria one at a time and finding out whether the procedures can be discerned from each other in terms of the remaining criteria.

Therefore, if we start from Table 13.2, by eliminating criterion A we get Table 13.3.

Table 13.3. Elimination of criterion A from Table 13.2

Voting procedure	Criteria					
	B	C	D	E	F	G
Amendment	1	1	1	0	0	0
Copeland	1	1	1	1	0	0
Dodgson	0	1	0	1	0	0
Max-min	0	1	1	1	0	0
Plurality	0	1	1	1	1	0
Borda	1	0	1	1	1	0
Approval	0	0	1	0	1	1
Runoff	1	1	0	1	0	0
Nanson	1	1	0	1	0	0

The two last rows of Table 13.3 are identical, so that to distinguish those two last rows, that is, Runoff and Nanson, criterion A is necessary, that is, criterion A is indispensable.

Then, we delete criterion B and obtain Table 13.4.

Table 13.4. Elimination of criterion B from Table 13.2

Voting procedure	Criteria					
	A	C	D	E	F	G
Amendment	1	1	1	0	0	0
Copeland	1	1	1	1	0	0
Dodgson	1	1	0	1	0	0
Max-min	1	1	1	1	0	0
Plurality	0	1	1	1	1	0
Borda	0	0	1	1	1	0
Approval	0	0	1	0	1	1
Runoff	0	1	0	1	0	0
Nanson	1	1	0	1	0	0

Now, Copeland and Max-Min become indistinguishable so that criterion B is indispensable.

Next, the elimination of criterion C leads to Table 13.5.

Table 13.5. Elimination of criterion C from Table 13.2

Voting procedure	Criteria					
	A	B	D	E	F	G
Amendment	1	1	1	0	0	0
Copeland	1	1	1	1	0	0
Dodgson	1	0	0	1	0	0
Max-min	1	0	1	1	0	0
Plurality	0	0	1	1	1	0
Borda	0	1	1	1	1	0
Approval	0	0	1	0	1	1
Runoff	0	1	0	1	0	0
Nanson	1	1	0	1	0	0

All rows in Table 13.5 are different. This means that criterion C is unnecessary to differentiate between those choice functions. Therefore, we can conclude that C is dispensable.

Further, we delete criterion D and obtain Table 13.6. Now, Copeland and Nanson are indistinguishable so that criterion D is indispensable.

Next, we eliminate criterion E and get Table 13.7.

Two uppermost rows are now identical. Thus, criterion E is indispensable.

Table 13.6. Elimination of criterion D from Table 13.6

Voting procedure	Criteria					
	A	B	C	E	F	G
Amendment	1	1	1	0	0	0
Copeland	1	1	1	1	0	0
Dodgson	1	0	1	1	0	0
Max-min	1	0	1	1	0	0
Plurality	0	0	1	1	1	0
Borda	0	1	0	1	1	0
Approval	0	0	0	0	1	1
Runoff	0	1	1	1	0	0
Nanson	1	1	1	1	0	0

Table 13.7. Elimination of criterion E from Table 13.2

Voting procedure	Criteria					
	A	B	C	D	F	G
Amendment	1	1	1	1	0	0
Copeland	1	1	1	1	0	0
Dodgson	1	0	1	0	0	0
Max-min	1	0	1	1	0	0
Plurality	0	0	1	1	1	0
Borda	0	1	0	1	1	0
Approval	0	0	0	1	1	1
Runoff	0	1	1	0	0	0
Nanson	1	1	1	0	0	0

Next, criterion F is eliminated which is shown in Table 13.8 in which no pair of rows is identical which means that criterion F is dispensable.

Table 13.8. Elimination of criterion F from Table 13.2

Voting procedure	Criteria					
	A	B	C	D	E	G
Amendment	1	1	1	1	0	0
Copeland	1	1	1	1	1	0
Dodgson	1	0	1	0	1	0
Max-min	1	0	1	1	1	0
Plurality	0	0	1	1	1	0
Borda	0	1	0	1	1	0
Approval	0	0	0	1	0	1
Runoff	0	1	1	0	1	0
Nanson	1	1	1	0	1	0

Finally, criterion G is eliminated which is shown in Table 13.9. We can see that all rows are different so that we can conclude that criterion G is dispensable.

Table 13.9. Elimination of criterion G from Table 13.2

Voting procedure	Criteria					
	A	B	C	D	E	F
Amendment	1	1	1	1	0	0
Copeland	1	1	1	1	1	0
Dodgson	1	0	1	0	1	0
Max-min	1	0	1	1	1	0
Plurality	0	0	1	1	1	1
Borda	0	1	0	1	1	1
Approval	0	0	0	1	0	1
Runoff	0	1	1	0	1	0
Nanson	1	1	1	0	1	0

We proceed further with our analysis. We can see from our discussion that the core is the set of indispensable criteria, that is, {A, B, D, E}. Moreover, the reduct is in this case unique, which need not always be the case, and is also equal to the set {A, B, D, E}. That is, we need just that set of criteria to distinguish the particular voting procedures from each other in the setting (criteria) assumed.

Table 13.10. Satisfaction of the criteria belonging to the core by the particular voting procedures

Voting procedure	Criteria			
	A	B	D	E
Amendment	1	1	1	0
Copeland	1	1	1	1
Dodgson	1	0	0	1
Max-min	1	0	1	1
Plurality	0	0	1	1
Borda	0	1	1	1
Approval	0	0	1	0
Runoff	0	1	0	1
Nanson	1	1	0	1

We can proceed with our analysis by considering the reduct (or core). In Table 13.10 we present a visualization of which criteria are indispensable in the sense that if we do not take them into account, two or more rows (corresponding to the respective voting procedures) become indistinguishable. For example, without criterion E, Amendment and Copeland would be indistinguishable, without D,

Copeland and Nanson would be indistinguishable, without B, Copeland and Max-Min would be indistinguishable, etc.

Table 13.10 expresses the most crucial properties or criteria of the voting procedures in the sense that the information it conveys would be sufficient to restore all information given in the source Table 13.2 because in fact we have just been reducing step by step information volume without reducing information contents.

Therefore, if we wish to characterize the choice functions in an economical way, we could use the values of the criteria given in Table 13.10 and present the results as in Table 13.11 where the subscripts of the particular criteria stand for the values

Table 13.11. An economical characterization of the voting procedures shown in Table 13.10

$A_1B_1D_1E_0$	$\longrightarrow$ Amendment
$A_1B_1D_1E_1$	$\longrightarrow$ Copeland
$A_1B_0D_0E_1$	$\longrightarrow$ Dodgson
$A_1B_0D_1E_1$	$\longrightarrow$ Max-min
$A_0B_0D_1E_1$	$\longrightarrow$ Plurality
$A_0B_1D_1E_1$	$\longrightarrow$ Borda
$A_0B_0D_1E_0$	$\longrightarrow$ Approval
$A_0B_1D_0E_1$	$\longrightarrow$ Runoff
$A_1B_1D_0E_1$	$\longrightarrow$ Nanson

they take on. For instance, to most concisely characterize Amendment, A, B and D should be 1 while E should be 0, etc.

This is, however, not the most economic characterization. For clarity, we will not deal here with this, and refer the interested reader to Fedrizzi, Kacprzyk and Nurmi [10] to learn that the minimal (most economical) characterization of the voting procedures in terms of information given in Table 13.2 can be portrayed as shown in Table 13.12.

As we can notice, the most economic characterization of the voting procedures (shown in (13.12))– which is essentially equivalent to a rich yet redundant characterization presented in the source data set in Table 13.2 – is very compact.

13.5 Similarity and Distances between Voting Procedures

The analysis of similarity of the voting procedures presented in the previous section does provide much insight and information yet it is rather of a qualitative type. In this section we will extend it towards a quantitative analysis while still operating in what might be considered an indiscernibility type context employed in the previous section.

First, we will assume that we will operate on the characterization of the voting procedures as shown in Table 13.2, that is, just after the reduction of identical rows

Table 13.12. The minimal (most economical) characterization of the voting procedures shown in Table 13.10

A_1E_0	$\longrightarrow$ Amendment
$A_1B_1D_1E_1$	$\longrightarrow$ Copeland
B_0D_0	$\longrightarrow$ Dodgson
$A_1B_0D_1$	$\longrightarrow$ Max-min
$A_0B_0E_1$	$\longrightarrow$ Plurality
$A_0B_1D_1$	$\longrightarrow$ Borda
A_0E_0	$\longrightarrow$ Approval
A_0D_0	$\longrightarrow$ Runoff
$A_1B_1D_0$	$\longrightarrow$ Nanson

in Table 13.1 has been done, without the further reductions as shown in Tables 13.3 – 13.10. This will better serve the purpose, and we will leave the problem of similarity analyses on reduced representations, which is somehow specific, for our future works. Again, to show the essence, we will consider the specific example as the method can be generalized in a straightforward way.

Suppose that we start with information given in Table 13.2 which, for the convenience of the readers, will be shown again in Table 13.13.

Table 13.13. Satisfaction of 7 criteria by 9 equivalent (classes of) voting procedures as shown in Table 13.2

Voting procedure	Criteria						
	A	B	C	D	E	F	G
Amendment	1	1	1	1	0	0	0
Copeland	1	1	1	1	1	0	0
Dodgson	1	0	1	0	1	0	0
Max-min	1	0	1	1	1	0	0
Plurality	0	0	1	1	1	1	0
Borda	0	1	0	1	1	1	0
Approval	0	0	0	1	0	1	1
Runoff	0	1	1	0	1	0	0
Nanson	1	1	1	0	1	0	0

For each pair of voting procedures, $(x,y) \in V^2$, where V is the set of voting procedures (9 in our case, as in Table 13.2), and for each criterion z, $z \in Z$, where Z is the set of criteria assumed (cf. Table 13.2), we define the following function $v_z : V \times V \longrightarrow \{0,1\}$, such that

$$v_z(x,y) = \begin{cases} 1 \text{ if } x \text{ and } y \text{ take on the same values for criterion } z \\ 0 \text{ otherwise} \end{cases} \tag{13.4}$$

For example, for the data given in Table 13.2:

$$v_{\mathrm{A}}(Amendment, Copeland) = 1$$
$$v_{\mathrm{E}}(Amendment, Copeland) = 0$$

In the simplest way the agreement between two voting procedures, $x, y \in V$, denoted by $A(x,y)$, $A : V \times V \longrightarrow \{0, \ldots, \mathrm{card}\, Z\}$, can be defined in terms of $v_z(x,y)$ given by (13.4) as follows:

$$A(x,y) = \sum_{z \in Z} v_z(x,y) \tag{13.5}$$

We thus get the following matrix of agreements (Table 13.14):

Table 13.14. Values of agreements between the particular voting procedures due to (13.5)

	Voting procedure								
Voting procedure	Amendment	Copeland	Dodgson	Max-min	Plurality	Borda	Approval	Runoff	Nanson
Amendment	-	6	4	5	3	3	1	4	5
Copeland		-	5	6	4	4	1	5	6
Dodgson			-	6	4	2	2	5	6
Max-min				-	5	3	2	4	5
Plurality					-	5	4	4	3
Borda						-	4	4	3
Approval							-	1	0
Runoff								-	6
Nanson									-

Since the agreement between identical voting procedures does not matter, the diagonal entries are denoted "-", and since the agreement function is symmetric, only the upper half of the matrix is shown.

The normalized distance between two voting procedures $x, y \in V$, can be defined in a straightforward way as

$$D(x,y) = 1 - \frac{A(x,y)}{\mathrm{card} Z} \tag{13.6}$$

where $A(x,y)$ is given by (13.5) and card Z is the number of criteria.

Therefore, using (13.6), we obtain the matrix of normalized distances between the voting procedures given by Table 13.15.

One can notice that, not surprisingly, Copeland, Max-Min, Dodgson and Nanson form a group of voting procedures which are not more than two criteria away of each other. Quite closely related to that group are Runoff and Amendment. By the way, except for Runoff, all these procedures are the Condorcet extensions, that is, they result in the choice of the Condorcet winner if it exists.

The so-called positional methods, that is, Plurality, Borda and Approval, seem to be rather far away from the rest of the procedures. This holds particularly for Approval.

Table 13.15. Normalized distances between the particular voting procedures due to (13.6)

	Voting procedure								
Voting procedure	Amendment	Copeland	Dodgson	Max-min	Plurality	Borda	Approval	Runoff	Nanson
Amendment	-	$\frac{1}{7}$	$\frac{3}{7}$	$\frac{2}{7}$	$\frac{4}{7}$	$\frac{4}{7}$	$\frac{6}{7}$	$\frac{3}{7}$	$\frac{2}{7}$
Copeland		-	$\frac{2}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{6}{7}$	$\frac{2}{7}$	$\frac{1}{7}$
Dodgson			-	$\frac{1}{7}$	$\frac{3}{7}$	$\frac{5}{7}$	$\frac{5}{7}$	$\frac{2}{7}$	$\frac{1}{7}$
Max-min				-	$\frac{2}{7}$	$\frac{4}{7}$	$\frac{5}{7}$	$\frac{3}{7}$	$\frac{2}{7}$
Plurality					-	$\frac{2}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{4}{7}$
Borda						-	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{4}{7}$
Approval							-	$\frac{6}{7}$	$\frac{7}{7}$
Runoff								-	$\frac{1}{7}$
Nanson									-

The results obtained are informative and valuable. On the one hand, they confirm to some extent knowledge and intuition of the voting theorists, which speaks positively for the correctness and usefulness of the method. On the other hand, the method provides a simple and intuitively appealing apparatus to determine in a systematic way some classes of the voting procedures which share some common properties so that they can be considered to be equivalent.

We think that such an equivalence analysis can be important because one will be able to extend many analytic and experimental results obtained in the field for the particular voting procedures, but dealt with before as different ones, to other ones which have been found to be equivalent. Of course, one should bear in mind that all this is up to the basic assumption of the set of criteria against which the voting procedures are compared.

It is easy to see that the similarity measure employed in the above analysis is straightforward and simple. In practice, many more sophisticated similarity measures are used, and we will present here the use of one of the most relevant and widely employed one due to Jaccard-Needham.

The Jaccard-Needham measure of similarity of binary patterns can be presented in a simple way by using the setting of Tubbs [50] proposed in the field of pattern recognition.

A binary vector Z of dimension N is defined as:

$$Z = (z_1, z_2, \ldots, z_N) \tag{13.7}$$

where $z_i \in \{0,1\}, \forall i \in \{1,2,\ldots,N\}$.

The set of all N-dimensional binary vectors is denoted by Ω, the *unit binary vector*, $I \in \Omega$, is a binary vector such that $z_i = 1, \forall i \in \{1,2,\ldots,N\}$. The *complement* of a binary vector $Z \in \Omega$ is $\overline{Z} = I - Z$.

If we have two binary vectors, $X, Y \in \Omega$, then we denote by $S_{i,j}(X,Y)$ the number of matches of i in vector X and j in vector Y, $i, j \in \{0,1\}$. That is, if we have two vectors:

$$X = [0,1,1,0,1,0,0,1,1,0]$$
$$Y = [1,1,0,0,1,1,0,0,1,0]$$

then we have:

$$S_{00}(X,Y) = 3$$
$$S_{01}(X,Y) = 2$$
$$S_{10}(X,Y) = 2$$
$$S_{11}(X,Y) = 3$$

Formally, we can define those measures as follows. First, for vectors $X = (x_1, x_2, \ldots, x_N)$ and $Y = (y_1, y_2, \ldots, y_N)$

$$v_{ij}^k(X,Y) = \begin{cases} 1 & \text{if } x_k = i \text{ and } y_k = j \\ 0 & \text{otherwise} \end{cases} \tag{13.8}$$

then

$$S_{ij}(X,Y) = \sum_{k=1}^{N} v_{ij}^k(X,Y) \tag{13.9}$$

One can easily notice that

$$S_{00}(X,Y) = \overline{X} \times \overline{Y}^T \tag{13.10}$$
$$S_{11}(X,Y) = X \times Y^T \tag{13.11}$$

Then, following the notation of Tubbs [50], the S_{ij}'s, $i, j \in \{0,1\}$, can be used to define many well known measures of *similarity*, $S(X,Y)$. Specifically, the Jaccard–Needham measure is defined as:

$$S_{J-N} = \frac{S_{11}}{S_{11} + S_{10} + S_{01}} \tag{13.12}$$

First, as in the case of the indiscernibility based comparison method for voting procedures presented in the previous section, we will use as the point of departure the binary matrix given in Table 13.2 which shows the satisfaction (= 1) or a lack of satisfaction (= 0) of the A, B, C, D, E, F, G criteria by the 9 (classes of) voting procedures.

Now, we will calculate S_{ij}, $i, j \in \{0,1\}$, according to (13.8)–(13.9), for the particular pairs of 9 voting procedures which will be presented in Table 13.16 the entries of which are given as $[S_{00}, S_{01}, S_{10}, S_{11}]$ for each pair.

Then, due to (13.12), with the values of $[S_{00}, S_{01}, S_{10}, S_{11}]$ shown in Table 13.16, we can calculate the values of the Jaccard-Needham similarity measure as shown in Table 13.17.

Notice that the results of our normalized distance based similarity analysis of the voting procedures shown in Table 13.14 are clearly different than the results

Table 13.16. Values of $[S_{00}, S_{01}, S_{10}, S_{11}]$ for the particular pairs of the voting procedures

	Voting procedure								
Voting procedure	Amendment	Copeland	Dodgson	Max-min	Plurality	Borda	Approval	Runoff	Nanson
Amendment	-	[2,1,0,4]	[2,1,2,2]	[2,1,1,3]	[1,2,2,2]	[1,2,2,2]	[1,2,3,1]	[2,1,2,2]	[2,1,1,3]
Copeland		-	[2,0,2,3]	[2,0,1,4]	[1,1,2,3]	[1,1,2,3]	[0,2,4,1]	[2,0,2,3]	[2,0,1,4]
Dodgson			-	[3,1,0,3]	[2,2,1,2]	[1,3,2,1]	[1,3,2,1]	[2,2,2,1]	[3,1,0,3]
Max-min				-	[2,1,1,3]	[1,2,2,2]	[1,2,3,1]	[2,1,2,2]	[2,1,1,3]
Plurality					-	[2,1,1,3]	[2,1,2,2]	[2,1,2,2]	[1,2,2,2]
Borda						-	[2,1,2,2]	[2,1,2,2]	[1,2,2,2]
Approval							-	[1,3,3,0]	[0,4,3,0]
Runoff								-	[3,1,0,3]
Nanson									-

Table 13.17. Value of the degree of similarity for the particular pairs of voting procedures using the Jaccard–Needham measure of similarity (13.12)

	Voting procedure								
Voting procedure	Amendment	Copeland	Dodgson	Max-min	Plurality	Borda	Approval	Runoff	Nanson
Amendment	-	0.80	0.40	0.60	0.33	0.33	0.17	0.40	0.60
Copeland		-	0.60	0.80	0.50	0.50	0.14	0.60	0.80
Dodgson			-	0.75	0.40	0.17	0.00	0.50	0.75
Max-min				-	0.60	0.33	0.17	0.40	0.60
Plurality					-	0.60	0.40	0.40	0.33
Borda						-	0.40	0.40	0.33
Approval							-	0.00	0.00
Runoff								-	0.75
Nanson									-

of an explicit Jaccard-Needham similarity measure based analysis shown in Table 13.17, the very essence of both of them is close. One point should, however, be added. To wit, the Jaccard-Needham measure ignores those instances where neither of the compared entities has the property under investigation. Surely, among voting systems the fact that two systems fail on a given criterion is potentially an important aspect of their similarity. Thus, in this regard, the straight-forward normalized distance-based similarity gives more information about system similarity than the Jaccard-Needham measure.

13.6 Concluding Remarks

We have presented a method of analysis of the similarity of voting procedures aimed at finding some sets of similar (maybe equivalent) procedures assuming that these procedures are compared against a set of well established criteria. Using some extension of an idea to use for that purpose, as the first step, some intuitively appealing tools and techniques of rough sets theory, as proposed by Fedrizzi, Kacprzyk and Nurmi [10], we have first of all employed some recent findings in voting theory

which have suggested that some voting procedures and criteria should be considered as equivalent, we have modified our previous analysis and obtained a more compact characterization of the particular classes of the voting procedures and the criteria. This, through the use of rough sets tools and techniques, has resulted in a indiscernibility type qualitative analysis. It was then enhanced with a quantitative analysis based on a simple yet intuitively appealing normalized distance-based similarity measure and then on the Jaccard-Needham explicit similarity based measure. The results obtained provide much insight into the very differences and common properties of the particular voting procedures and can be useful both for formal analyses and practical applications, in human voting and multi-agent software systems. As for a further research, an extended approach based on the use of distance and similarity measures between binary patterns should provide valuable results. Moreover, a joint account for similarity and dissimilarity, which need not be dual in the case of some measures, can provide an additional insight.

References

1. Arrow, K.J.: A difficulty in the concept of social welfare. Journal of Political Economy 58(4), 328–346 (1950)
2. Arrow, K.J., Sen, A.K., Suzumura, K. (eds.): Handbook of Social Choice and Welfare, vol. 1. Elsevier, Amsterdam (2002)
3. Baigent, M.: Metric rationalisation of social choice functions according to principles of social choice. Mathematical Social Sciences 13(1), 59–65 (1987)
4. Diday, E., Esposito, F.: An introduction to symbolic data analysis and SODAS software. Journal of Intelligent Data Analysis 7, 583–602 (2003)
5. Elkind, E., Faliszewski, P., Slinko, A.: On the role of distances in defining voting rules. In: van der Hoek, W., Kaminka, G.A., Lespérance, Y., Luck, M., Sen, S. (eds.) Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, pp. 375–382. International Foundation for Autonomous Agents and Multiagent Systems (2010)
6. Esposito, F., Malerba, D., Appice, A.: Dissimilarity and matching. In: Diday, E., Noirhomme-Fraiture, M. (eds.) Symbolic Data Analysis and the SODAS Software, pp. 61–66. Wiley, Chichester (2008)
7. Esposito, F., Malerba, D., Lisi, F.A.: Matching symbolic objects. In: Bock, H.-H., Diday, E. (eds.) Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, pp. 186–197. Springer, Berlin (2000)
8. Esposito, F., Malerba, D., Tamma, V.: Dissimilarity measures for symbolic objects. In: Bock, H.-H., Diday, E. (eds.) Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, pp. 165–185. Springer, Berlin (2000)
9. Esposito, F., Malerba, D., Tamma, V., Bock, H.-H.: Classical similarity and dissimilarity measures. In: Bock, H.-H., Diday, E. (eds.) Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, pp. 139–152. Springer, Berlin (2000)
10. Fedrizzi, M., Kacprzyk, J., Nurmi, H.: How different are social choice functions: A rough sets approach. Quality and Quantity 30, 87–99 (1996)

11. Fishburn, P.C.: Condorcet social choice functions. SIAM Journal on Applied Mathematics 33, 469–489 (1977)
12. Gibbard, A.: Manipulation of voting schemes: A general result. Econometrica 41(4), 587–601 (1973)
13. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
14. Kacprzyk, J.: Group decision making with a fuzzy majority. Fuzzy Sets and Systems 18, 105–118 (1986)
15. Kacprzyk, J., Fedrizzi, M.: A 'human consistent' degree of consensus based on fuzzy logic with linguistic quantifiers. Mathematical Social Sciences 18, 275–290 (1989)
16. Kacprzyk, J., Fedrizzi, M., Nurmi, H.: Group decision making and consensus under fuzzy preferences and fuzzy majority. Fuzzy Sets and Systems 49, 21–31 (1992)
17. Kacprzyk, J., Zadrożny, S.: Towards a general and unified characterization of individual and collective choice functions under fuzzy and nonfuzzy preferences and majority via the ordered weighted average operators. International Journal of Intelligent Systems 24(1), 4–26 (2009)
18. Kacprzyk, J., Zadrożny, S.: Towards human consistent data driven decision support systems using verbalization of data mining results via linguistic data summaries. Bulletin of the Polish Academy of Sciences: Technical Sciences 58(3), 359–370 (2010)
19. Kelly, J.S.: Arrow Impossibility Theorems. Academic Press, New York (1978)
20. Kelly, J.S.: Social Choice Theory. Springer, Berlin (1988)
21. Malerba, D., Sanarico, L., Tamma, V.: A comparison of dissimilarity measures for Boolean symbolic data. In: Brito, P., Costa, J., Malerba, D. (eds.) Proceedings of the ECML 2000/MLnet Workshop on Dealing with Structured Data in Machine Learning and Statistics, Barcelona, Spain, pp. 44–52 (2000)
22. May, K.: A set of independent, necessary and sufficient conditions for simple majority decision. Econometrica 20(4), 680–684 (1952)
23. McCabe-Dansted, J.C., Slinko, A.: Exploratory Analysis of Similarities Between Social Choice Rules. Group Decision and Negotiation 15(1), 77–107 (2006)
24. Merrill, S.: Making Multicandidate Elections More Democratic. Princeton University Press, Princeton (1988)
25. Moulin, H.: The Strategy of Social Choice. In: Advanced Textbooks in Economics. North-Holland, Amsterdam (1983)
26. Moulin, H.: Axioms of Cooperative Decision Making. Cambridge University Press, Cambridge (1991)
27. Nurmi, H.: Comparing Voting Systems. Reidel, Dordrecht (1987)
28. Nurmi, H.: Discrepancies in the outcomes resulting from different voting schemes. Theory and Decision 25, 193–208 (1988)
29. Nurmi, H.: Inferential modes in applying social choice theory. In: Munier, B.R., Shakun, M.F. (eds.) Compromise, Negotiation and Group Decision. Reidel, Dordrecht (1988)
30. Nurmi, H.: An assessment of voting system simulations. Public Choice 73, 459–487 (1992)
31. Nurmi, H., Kacprzyk, J.: On fuzzy tournaments and their solution concepts in group decision making. European Journal of Operational Research 51, 223–232 (1991)
32. Nurmi, H., Kacprzyk, J., Fedrizzi, M.: Probabilistic, fuzzy and rough concepts in social choice. European Journal of Operational Research 95, 264–277 (1996)
33. Pawlak, Z.: Rough sets. International Journal of Information and Computer Sciences 11, 341–356 (1982)
34. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)

35. Pawlak, Z.: An inquiry into anatomy of conflicts. Information Sciences 109, 65–78 (1998)
36. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
37. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177(1), 28–40 (2007)
38. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177, 41–73 (2007)
39. Peters, J.F., Skowron, A., Stepaniuk, J.: Rough sets: Foundations and perspectives. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7787–7797. Springer, Berlin (2009)
40. Pitt, J., Kamara, L., Sergot, M., Artikis, A.: Formalization of a voting protocol for virtual organizations. In: Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M., Wooldridge, M. (eds.) Proc. 4th AAMAS 2005, pp. 373–380. ACM (2005)
41. Pitt, J., Kamara, L., Sergot, M., Artikis, A.: Voting in online deliberative assemblies. In: Gardner, A., Sartor, G. (eds.) Proc. 10th ICAIL, pp. 195–204. ACM (2005)
42. Pitt, J., Kamara, L., Sergot, M., Artikis, A.: Voting in multi-agent systems. The Computer Journal 49(2), 156–170 (2006)
43. Plott, C.R.: Axiomatic social choice theory: an overview and interpretation. American Journal of Political Science 20, 511–596 (1976)
44. Polkowski, L.: A set theory for rough sets. Toward a formal calculus of vague statements. Fundamenta Informaticae 71(1), 49–61 (2006)
45. Richelson, J.: A comparative analysis of social choice functions I, II, III: A summary. Behavioral Science 24, 355 (1979)
46. Riker, W.H.: Liberalism against Populism. W.H. Freeman, San Francisco (1982)
47. Satterthwaite, M.A.: Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. Journal of Economic Theory 10, 187–217 (1975)
48. Schwartz, T.: The Logic of Collective Choice. Columbia University Press, New York (1986)
49. Straffin, P.D.: Topics in the Theory of Voting. Birkhäuser, Boston (1980)
50. Tubbs, J.D.: A note on binary template matching. Pattern Recognition 22(4), 359–365 (1989)
51. Yager, R.R., Kacprzyk, J. (eds.): The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Boston (1997)
52. Yager, R.R., Kacprzyk, J., Beliakov, G. (eds.): Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice. Springer (2011)
53. Yao, Y.Y., Zhao, Y.: Discernibility matrix simplification for constructing attribute reducts. Information Sciences 179, 867–882 (2009)

Chapter 14
Algebras for Information Systems*

Md. Aquil Khan** and Mohua Banerjee

Abstract. We present algebraic formalisms for different kinds of information systems, viz. deterministic, incomplete, and non-deterministic. Algebraic structures generated from these information systems are considered and corresponding abstract algebras are proposed. Representation theorems for these classes of abstract algebras are proved, which lead us to equational logics for deterministic, incomplete, and non-deterministic information systems.

Keywords: Information system, indiscernibility relation, similarity relation, Boolean algebra with operators.

Md. Aquil Khan
The Institute of Mathematical Sciences
Chennai 600113, India
e-mail: mdaquilkhan@gmail.com
and
Institute of Mathematics, The University of Warsaw
2, Banacha St., Warsaw 02-097, Poland

Mohua Banerjee
Indian Institute of Technology
Kanpur 208016, India
e-mail: mohua@iitk.ac.in

* A part of this chapter appeared in preliminary form in the proceedings of the conference PReMI 2011.

** This work was finalized when the first author was affiliated to the Fraunhofer Institute for Secure Information Technology SIT, Darmstadt, Germany, during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 381–407.
springerlink.com

14.1 Introduction

With time, Pawlak's simple rough set model has seen many generalizations due to demands from different practical situations (*e.g.,* [12, 39, 32, 22, 33, 34]). As we know, the notion of an *approximation space* [29], viz. a tuple (U,R), where U is a non-empty set and R an equivalence relation, plays a crucial role in Pawlak's rough set theory. A useful natural generalization is where the relation R is not necessarily an equivalence. For instance, in [33, 17], a *tolerance approximation space* is considered, where R is a tolerance relation. The notion of lower and upper approximations of a set in these generalized approximation spaces is then defined in a natural way. In Pawlak's definition of lower and upper approximations of a subset X of the domain U in an approximation space (U,R), equivalence classes $[x]_R$ of objects are replaced by the set $R(x) := \{y \in U : (x,y) \in R\}$. That is, lower and upper approximations of a set $X(\subseteq U)$ in a generalized approximation space (U,R) are given as:

$$\underline{X}_R := \{x \in U : R(x) \subseteq X\}, \text{ and } \overline{X}_R := \{x \in U : R(x) \cap X \neq \emptyset\}.$$

There is another way to look at generalizations of Pawlak's rough set theory, viz. from the point of view of *information systems*. Most applications of rough set theory are based on these attribute-value representation models.

Definition 14.1. A *deterministic information system* (DIS) $\mathcal{S} := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f)$, comprises a non-empty set U of objects, $\mathcal{A}$ of attributes, $\mathcal{V}_a$ of attribute values for each $a \in \mathcal{A}$, and information function $f : U \times \mathcal{A} \to \bigcup_{a\in\mathcal{A}} \mathcal{V}_a$ such that $f(x,a) \in \mathcal{V}_a$.

$\mathcal{S}_1$ and $\mathcal{S}_2$ of Table 14.1, which provide information about three patients $P1 - P3$ regarding attributes "Temperature (T)", and "Headache (H)", are examples of DISs.

Table 14.1. DISs $\mathcal{S}_1$ ans $\mathcal{S}_2$

(a) DIS $\mathcal{S}_1$

Patient	T	H
P1	very high	yes
P2	very high	no
P3	high	yes

(b) DIS $\mathcal{S}_2$

Patient	T	H
P1	very high	yes
P2	very high	no
P3	no	yes

Given a deterministic information system $\mathcal{S} := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f)$ and a set $B \subseteq \mathcal{A}$, the *indiscernibility relation* $Ind_B^{\mathcal{S}}$ is an equivalence relation on U defined by:

$$(x,y) \in Ind_B^{\mathcal{S}}, \text{ if and only if } f(x,a) = f(y,a) \text{ for all } a \in B.$$

Thus, given a DIS $\mathcal{S}$ and a set B of attributes, we obtain an approximation space $(U, Ind_B^{\mathcal{S}})$. For instance, in the above example, corresponding to the attribute set

$B := \{\text{T}, \text{H}\}$, DISs $\mathcal{S}_1$ and $\mathcal{S}_2$ give rise to the approximation spaces $(\{\text{P1}, \text{P2}, \text{P3}\}, Ind_B^{\mathcal{S}_1})$ and $(\{\text{P1}, \text{P2}, \text{P3}\}, Ind_B^{\mathcal{S}_2})$ respectively, where

$$Ind_B^{\mathcal{S}_1} = Ind_B^{\mathcal{S}_2} = \{(\text{P1}, \text{P1}), (\text{P2}, \text{P2}), (\text{P3}, \text{P3})\}.$$

One may then approximate elements of a set X with respect to an attribute set B using the notion of lower/upper approximations in the approximation space $(U, Ind_B^{\mathcal{S}})$. Note that we may have two different DISs $\mathcal{K}_1 := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f_1)$ and $\mathcal{K}_2 := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f_2)$ with the same domain and the same sets of attribute, attribute-values, such that $Ind_{\mathcal{A}}^{\mathcal{K}_1} = Ind_{\mathcal{A}}^{\mathcal{K}_2}$, hence generating the same approximation space with respect to the attribute set $\mathcal{A}$. This is the case, for example, with the DISs of Table 14.1.

The notion of a deterministic information system has been generalized in many ways to consider different practical situations. For instance, information regarding values of some attribute for some object may not be available (unlike the case of a DIS, where the information is *complete*). A distinguished attribute-value $*$ is used to depict such a situation.

Definition 14.2. A tuple $\mathcal{S} := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f)$ is called an *information system* (IS), where $U, \mathcal{A}, Val_a, f$ are as in Definition 14.1 and $* \in \bigcap_{a\in\mathcal{A}} Val_a$. An information system which satisfies $f(x,a) = *$ for some $x \in U$ and $a \in \mathcal{A}$ will be called an *incomplete information system* (IIS).

Observe that a deterministic information system can be identified with the information system $\mathcal{S} := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f)$, where $f(x,a) \neq *$ for all $x \in U$ and $a \in \mathcal{A}$.

In [20, 21], instead of an indiscernibility relation, a *similarity* relation (defined below) is considered as the distinguishability relation in the context of an incomplete information system. The assumption here is that the real value of missing attributes is one from the attribute domain.
$(x,y) \in Sim_B^{\mathcal{S}}$ if and only if , $f(x,a) = f(y,a)$ or $f(x,a) = *$, or $f(y,a) = *$, for all $a \in B$.
One could easily verify that $Sim_B^{\mathcal{S}}$ is a tolerance relation, and thus, an IIS $\mathcal{S}$ and an attribute set B give rise to a tolerance approximation space $(U, Sim_B^{\mathcal{S}})$.

DISs are deterministic in the sense that objects take a single value for each attribute. Thus, a natural generalization of DISs is obtained by allowing an object to take a *set of values* for an attribute.

Definition 14.3. A tuple $\mathcal{S} := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f)$ is called a *non-deterministic information system* (NIS), where $U, \mathcal{A}, Val_a$ are as in Definition 14.1 and $f : U \times \mathcal{A} \to \wp(\bigcup_{a\in\mathcal{A}} \mathcal{V}_a)$ such that $f(x,a) \subseteq \mathcal{V}_a$.

Note that an indiscernibility relation $Ind_B^{\mathcal{S}}$ for NISs can be defined in a way identical to that for DISs.

One may attach different interpretations with "$f(x,a) = V$". For instance, as exemplified in [9, 10], if a is the attribute "speaking a language", then $f(x,a) = \{\text{German}, \text{English}\}$ can be interpreted as (i) x speaks German and English and no

other languages, (ii) x speaks German and English and possibly other languages, (iii) x speaks German or English but not both, or (iv) x speaks German or English or both. Motivated by these interpretations several relations apart from the indiscernibility relation are defined on non-deterministic information systems (*e.g.,* [28, 36, 10]). We list a few of them below.

Similarity $(x,y) \in Sim_B^S$ if and only if $f(x,a) \cap f(y,a) \neq \emptyset$ for all $a \in B$.
Inclusion $(x,y) \in In_B^S$ if and only if $f(x,a) \subseteq f(y,a)$ for all $a \in B$.
Negative similarity $(x,y) \in NSim_B^S$ if and only if $\sim f(x,a) \cap \sim f(y,a) \neq \emptyset$ for all $a \in B$, where $\sim$ is the complementation relative to $\mathcal{V}_a$.
Complementarity $(x,y) \in Com_B^S$ if and only if $f(x,a) = \sim f(y,a)$ for all $a \in B$.
Weak indiscernibility $(x,y) \in wInd_B^S$ if and only if $f(x,a) = f(y,a)$ for some $a \in B$.
Weak similarity $(x,y) \in wSim_B^S$ if and only if $f(x,a) \cap f(y,a) \neq \emptyset$ for some $a \in B$.
Weak inclusion $(x,y) \in wIn_B^S$ if and only if $f(x,a) \subseteq f(y,a)$ for some $a \in B$.
Weak negative similarity $(x,y) \in wNSim_B^S$ if and only if $\sim f(x,a) \cap \sim f(y,a) \neq \emptyset$ for some $a \in B$.
Weak complementarity $(x,y) \in wCom_B^S$ if and only if $f(x,a) = \sim f(y,a)$ for some $a \in B$.

Each of the relations defined above gives rise to a generalized approximation space, where the relation may not be an equivalence. Thus, one can approximate any subset of the domain using the lower and upper approximations defined on these generalized approximation spaces.

14.1.1 Towards an Algebra for Information Systems

In this chapter, we study classes of algebraic structures that are obtained from deterministic, incomplete and non-deterministic information systems. An algebraic approach to rough set theory was first presented by Iwiński in 1987 [14]. Since then, substantial work has been done on algebraic aspects of the theory (*e.g.,* cf. [17, 31, 4, 37]). In one direction, different representations of rough sets have been considered, and endowed with algebraic structures. It is observed that the algebras induced from approximation spaces are instances of various known as well as new algebraic structures, such as quasi-Boolean algebras, double Stone algebras, Nelson algebras, Łukasiewicz algebras, and topological quasi-Boolean algebras. A detailed survey can be found in [4]. In another direction of research, lower/upper approximations are viewed as unary operators mapping a set to its lower/upper approximations. This observation leads to a class of Boolean algebras with operators (BAO). For instance, in [38], a BAO consisting of two unary operators L and H is considered, where these operators are used to capture the lower and upper approximations. We would like to mention here that the motivation of such a Boolean algebra with operators comes from approximation operators induced by approximation spaces, where

the attribute set does not come into the picture. However, in ISs, as evident, the notions of approximations are not absolute, but *relative to attribute sets*. In fact, a DIS $\mathcal{S} := (U, \mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a, f)$ determines an algebra $\mathcal{B}_\mathcal{S} := (\wp(U), \cap, \sim, \emptyset, \{\overline{Ind_B^\mathcal{S}}\}_{B\subseteq\mathcal{A}})$, where $\emptyset$ denotes the empty set, $\sim$ is the operation of taking the complement of a set relative to U, $\cap$ that of taking the intersection of two sets and $\overline{Ind_B^\mathcal{S}}$ is a unary operator on $\wp(U)$ mapping a set $X(\subseteq U)$ to $\overline{X}_{Ind_B^\mathcal{S}}$. In [8], $\mathcal{B}_\mathcal{S}$ is called a *knowledge approximation algebra of type $\mathcal{A}$ derived from the DIS $\mathcal{S}$*.

In this chapter, we are interested in the following standard line of investigation in algebraic studies of classes of structures. In order to study a class $\mathcal{C}$ of structures obtained from information systems, one tries to abstract it through a class $\mathcal{F}$ of structures given by a set of axioms, such that each member of $\mathcal{C}$ is also a member of $\mathcal{F}$. Moreover, the adequacy of the class $\mathcal{F}$ of abstract structures for $\mathcal{C}$ is proved through a *representation theorem*, which involves showing that for every abstract structure $\mathfrak{A} \in \mathcal{F}$, there is a structure $\mathfrak{C} \in \mathcal{C}$, and an isomorphism/embedding (in an appropriate sense) from $\mathfrak{A}$ to $\mathfrak{C}$. In the context of frames, this is just the notion of *informational representability* [27]. It may be mentioned that informationally representable frames were first studied in [35], and a detailed study of informational representability can be found in [11].

In the above lines, the following abstract algebra is proposed in [8] corresponding to knowledge approximation algebras derived from DISs (with finite attribute set $\mathcal{A}$).

Definition 14.4. A structure $\mathfrak{B} := (\mathcal{B}, \kappa_P)_{P\subseteq\mathcal{A}}$ is a *knowledge approximation algebra of type $\mathcal{A}$* (a finite set), if $\kappa_P \in B^B$ for each $P \subseteq \mathcal{A}$ and the following axioms hold for all $x, y \in B$ and $P, Q \subseteq \mathcal{A}$.

(A_0) $\mathcal{B} := (B, \vee, \neg, 0)$ is a complete atomic Boolean algebra.
(A_1) $\kappa_P 0 = 0$.
(A_2) $\kappa_P x \geq x$.
(A_3) $\kappa_P(x \wedge \kappa_P y) = \kappa_P x \wedge \kappa_P y$.
(A_4) $x \neq 0$ implies $\kappa_\emptyset x = 1$.
(A_5) $\kappa_{P\cup Q} x = (\kappa_P x) \wedge (\kappa_Q x)$ if x is an atom of $\mathcal{B}$.

A representation theorem is also presented in [8] stating that every knowledge approximation algebra of type $\mathcal{A}$ is isomorphic to a knowledge approximation algebra of type $\mathcal{A}$ derived from some DIS. In order to see how the required DIS is generated from the given knowledge approximation algebra, we need a few definitions.

Given a knowledge approximation algebra $\mathfrak{B} := (\mathcal{B}, \kappa_P)_{P\subseteq\mathcal{A}}$, the *atomic structure* $\mathbf{At}(\mathbf{B})$ of $\mathfrak{B}$ is $(At(B), T_P)_{P\subseteq\mathcal{A}}$, where $At(B)$ is the set of atoms of B and for each $P \subseteq \mathcal{A}$, T_P is the equivalence relation on $At(B)$ such that

$$(x, y) \in T_P \text{ if and only if } y \leq \kappa_P x.$$

Now the atomic structure $\mathbf{At}(\mathbf{B})$, in turn, determines a DIS $\mathcal{S}(\mathbf{At}(\mathbf{B})) := (At(B), \mathcal{A}, \mathcal{V}, f)$, where $\mathcal{V} := \bigcup_{a\in\mathcal{A}} At(B)/T_{\{a\}}$ and $f(x, a) := [x]_{T_{\{a\}}}$. Finally, in the representation theorem it is shown that $\mathfrak{B} \cong \mathcal{B}_{\mathcal{S}(\mathbf{At}(\mathbf{B}))}$. Note that in this representation theorem, the atomicity property of $\mathfrak{B}$ plays a crucial role.

We note that Comer's work in [8] is confined to DISs only and does not talk about possible extensions to other types of information systems, such as incomplete and non-deterministic information systems. Moreover, the knowledge approximation algebra derived from a DIS $\mathcal{S}$ does not give a complete description of the DIS. In fact, attribute and attribute-value pairs, which are the main ingredients of a DIS, do not appear in this description. In the next section, we shall propose an algebraic formalism of DISs which captures this aspect. An abstract algebra for DISs will be proposed and the corresponding representation theorem will be proved in Section 14.3. In Section 14.4, we will see that this representation theorem also leads us to logics for DISs. In Sections 14.5, 14.6 and 14.7, we shall extend this formalism to incomplete and non-deterministic information systems as well. Section 14.8 concludes the chapter.

14.2 Algebra for Deterministic Information Systems

Let us consider the deterministic information systems $\mathcal{S}_1$ and $\mathcal{S}_2$ of Table 14.1. Table 14.2 below gives the lower and upper approximations of all the subsets of U with respect to indiscernibility relations corresponding to different sets of attributes. Observe that the two different DISs $\mathcal{S}_1$ and $\mathcal{S}_2$ generate the same knowledge approximation algebra $(\wp(U), \cap, \sim, \emptyset, \{\overline{Ind_B^{\mathcal{S}_i}}\}_{B\subseteq\mathcal{A}})$, where $\mathcal{A} := \{T, H\}$ and the operator $\overline{Ind_B^{\mathcal{S}_i}}$ is determined by Table 14.2. This fact shows that a knowledge approximation algebra does not give a complete description of the DISs. This observation leads us to the proposal of a *deterministic information system algebra* given as follows.

Table 14.2. Lower and upper approximations in the information systems $\mathcal{S}_i$, $i = 1, 2$

X	$\underline{X}_{Ind^{\mathcal{S}_i}_{\{T\}}}$	$\overline{X}_{Ind^{\mathcal{S}_i}_{\{T\}}}$	$\underline{X}_{Ind^{\mathcal{S}_i}_{\{H\}}}$	$\overline{X}_{Ind^{\mathcal{S}_i}_{\{H\}}}$	$\underline{X}_{Ind^{\mathcal{S}_i}_{\{T,H\}}}$	$\overline{X}_{Ind^{\mathcal{S}_i}_{\{T,H\}}}$
$\{P1\}$	$\emptyset$	$\{P1, P2\}$	$\emptyset$	$\{P1, P3\}$	$\{P1\}$	$\{P1\}$
$\{P2\}$	$\emptyset$	$\{P1, P2\}$	$\{P2\}$	$\{P2\}$	$\{P2\}$	$\{P2\}$
$\{P3\}$	$\{P3\}$	$\{P3\}$	$\emptyset$	$\{P1, P3\}$	$\{P3\}$	$\{P3\}$
$\{P1, P2\}$	$\{P1, P2\}$	$\{P1, P2\}$	$\emptyset$	U	$\{P1, P2\}$	$\{P1, P2\}$
$\{P1, P3\}$	$\emptyset$	U	$\{P1, P3\}$	$\{P1, P3\}$	$\{P1, P3\}$	$\{P1, P3\}$
$\{P2, P3\}$	$\{P3\}$	U	$\emptyset$	U	$\{P2, P3\}$	$\{P2, P3\}$
U	U	U	U	U	U	U
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Let us fix finite sets $\mathcal{A}$ of attributes and $\mathcal{V} := \bigcup_{a\in\mathcal{A}} \mathcal{V}_a$ of attribute values. Let $\mathcal{D}$ denote the set of all *descriptors* [30], viz. pairs (a, v), for each $a \in \mathcal{A}$, $v \in \mathcal{V}_a$. As we have already seen, given a deterministic information system $\mathcal{S} := (U, \mathcal{A}, \mathcal{V}, f)$, the upper approximations with respect to the indiscernibility relations $Ind_B^{\mathcal{S}}$, $B \subseteq \mathcal{A}$, determine unary operations $\overline{Ind_B^{\mathcal{S}}}$ on $\wp(U)$, viz.

$$\overline{Ind_B^{\mathcal{S}}}(X) := \overline{X}_{Ind_B^{\mathcal{S}}},\ X \subseteq U.$$

Similarly, one has the unary operations $\underline{Ind_B^{\mathcal{S}}}$ determined by lower approximations. Each descriptor (a,v) also determines a nullary operation (constant) $c^{\mathcal{S}}_{(a,v)}$ on $\wp(U)$:

$$c^{\mathcal{S}}_{(a,v)} := \{x \in U : f(x,a) = v\}.$$

Thus, we have the following definition. Let Ω be the tuple $(\mathcal{A}, \mathcal{V})$.

Definition 14.5. Let $\mathcal{S} := (U, \mathcal{A}, \mathcal{V}, f)$ be a deterministic information system. A *deterministic information system algebra (in brief,* DIS-algebra*) of type* Ω *generated by the deterministic information system* $\mathcal{S}$ is the structure

$$\mathcal{S}^* := (\wp(U), \cap, \sim, \emptyset, \{\underline{Ind_B^{\mathcal{S}}}\}_{B \subseteq \mathcal{A}}, \{c^{\mathcal{S}}_{\gamma}\}_{\gamma \in \mathcal{D}}).$$

Observe that a DIS-algebra generated by a DIS $\mathcal{S}$ is actually an extension of the knowledge approximation algebra derived from $\mathcal{S}$ with a collection of nullary operations. The DIS-algebra generated by the DIS $\mathcal{S}_i$, $i = 1,2$ (cf. Table 14.1) is given by $\mathcal{S}_i{}^* := (\wp(U),\ \cap,\ \sim,\ \emptyset,\ \{\underline{Ind_B^{\mathcal{S}_i}}\}_{B \subseteq \mathcal{A}},\ \{c^{\mathcal{S}_i}_{\gamma}\}_{\gamma \in \mathcal{D}})$, where $\mathcal{A} := \{T, H\}$, $\mathcal{D} := \{$(T, very high), (T,high), (T,no), (H,yes), (H,no)$\}$, $\underline{Ind_B^{\mathcal{S}_i}}$ is given by Table 14.2, and $c^{\mathcal{S}_i}_{\gamma}$ by Table 14.3 below. As expected, $\mathcal{S}_1{}^*$ and $\mathcal{S}_2{}^*$ differ only with respect to nullary operators.

Table 14.3. Nullary operators $c^{\mathcal{S}_i}_{\gamma}$, $i = 1,2$

γ	$c^{\mathcal{S}_1}_{\gamma}$	$c^{\mathcal{S}_2}_{\gamma}$
(T,very high)	$\{P1, P2\}$	$\{P1, P2\}$
(T, high)	$\{P3\}$	$\emptyset$
(T,no)	$\emptyset$	$\{P3\}$
(H,no)	$\{P2\}$	$\{P2\}$
(H,yes)	$\{P1, P3\}$	$\{P1, P3\}$

Notation 1. *For the elements x and y of a Boolean algebra* $(U, \wedge, \neg, 0)$*, we shall write* $x \to y$ *and* $x \leftrightarrow y$ *to denote the elements* $\neg x \vee y$ *and* $(x \to y) \wedge (y \to x)$*, respectively. Thus, in particular, for subsets X and Y of U in the power set Boolean algebra with domain* $\wp(U)$*,* $X \to Y$ *and* $X \leftrightarrow Y$*, respectively, represent the sets* $\sim X \cup Y$ *and* $(X \to Y) \cap (Y \to X)$*.*

The following proposition lists a few properties of DIS-algebras.

Proposition 14.1.

1. $\underline{Ind_B^{\mathcal{S}}}(X \cap Y) = \underline{Ind_B^{\mathcal{S}}}(X) \cap \underline{Ind_B^{\mathcal{S}}}(Y)$, $X \subseteq U$.
2. $\bigcup_{v \in \mathcal{V}_a} c^{\mathcal{S}}_{(a,v)} = U$.

3. $c^{\mathcal{S}}_{(a,v)} \cap c^{\mathcal{S}}_{(a,u)} = \emptyset$ *when* $v \neq u$.
4. $\underline{Ind^{\mathcal{S}}_C}(X) \subseteq \underline{Ind^{\mathcal{S}}_B}(X)$ *for* $C \subseteq B \subseteq \mathcal{A}$, *and* $X \subseteq U$.
5. $c^{\mathcal{S}}_{(a,v)} \subseteq \underline{Ind^{\mathcal{S}}_{\{a\}}}(c^{\mathcal{S}}_{(a,v)})$.
6. $c^{\mathcal{S}}_{(b,v)} \cap \underline{Ind^{\mathcal{S}}_{B\cup\{b\}}}(X) \subseteq \underline{Ind^{\mathcal{S}}_B}(c^{\mathcal{S}}_{(b,v)} \rightarrow X)$, $X \subseteq U$.
7. $\underline{Ind^{\mathcal{S}}_\emptyset}(X) \neq \emptyset$ *implies* $X = U$.
8. $\underline{Ind^{\mathcal{S}}_\emptyset}(U) = U$.

Proof. (1) It is enough to show that $\underline{X \cap Y}_{Ind^{\mathcal{S}}_B} = \underline{X}_{Ind^{\mathcal{S}}_B} \cap \underline{Y}_{Ind^{\mathcal{S}}_B}$. Here,

$$
\begin{aligned}
& x \in \underline{X \cap Y}_{Ind^{\mathcal{S}}_B} \\
\Longleftrightarrow\ & [x]_{Ind^{\mathcal{S}}_B} \subseteq X \cap Y \\
\Longleftrightarrow\ & [x]_{Ind^{\mathcal{S}}_B} \subseteq X, \text{ and } [x]_{Ind^{\mathcal{S}}_B} \subseteq Y \\
\Longleftrightarrow\ & x \in \underline{X}_{Ind^{\mathcal{S}}_B} \cap \underline{Y}_{Ind^{\mathcal{S}}_B}.
\end{aligned}
$$

(2) We just need to prove the inclusion $U \subseteq \bigcup_{v\in\mathcal{V}_a} c^{\mathcal{S}}_{(a,v)}$. So, let us take an arbitrary $x \in U$. Then, $f(x,a) = v$ for some $v \in \mathcal{V}_a$, and hence, we obtain $x \in c^{\mathcal{S}}_{(a,v)}$.
(3) Follows directly from the definition of the operators $c^{\mathcal{S}}_{(a,v)}$, and the fact that f is a function with domain $U \times \mathcal{A}$.
(4) From the definition of the indiscernibility relation, we obtain for $C \subseteq B$, $Ind^{\mathcal{S}}_B \subseteq Ind^{\mathcal{S}}_C$, and hence $\underline{X}_{Ind^{\mathcal{S}}_C} \subseteq \underline{X}_{Ind^{\mathcal{S}}_B}$. Now using the definition of the operators $\underline{Ind^{\mathcal{S}}_B}$, and $\underline{Ind^{\mathcal{S}}_C}$, we obtain the desired result.
(5) Let $x \in c^{\mathcal{S}}_{(a,v)}$. Then, we obtain $f(x,a) = v$. Now let us consider an arbitrary y such that $(x,y) \in Ind^{\mathcal{S}}_{\{a\}}$. Then, $f(y,a) = f(x,a) = v$, and hence, $y \in c^{\mathcal{S}}_{(a,v)}$. Thus, we have shown $x \in \underline{Ind^{\mathcal{S}}_{\{a\}}}(c^{\mathcal{S}}_{(a,v)})$.
(6) Let $x \in c^{\mathcal{S}}_{(b,v)} \cap \underline{Ind^{\mathcal{S}}_{B\cup\{b\}}}(X)$. Let us take an arbitrary y such that $(x,y) \in Ind^{\mathcal{S}}_B$. We need to show $y \in c^{\mathcal{S}}_{(b,v)} \rightarrow X$, that is, $y \in \sim c^{\mathcal{S}}_{(b,v)} \cup X$. Let us assume that $y \in c^{\mathcal{S}}_{(b,v)}$, we prove $y \in X$. Note that $x, y \in c^{\mathcal{S}}_{(b,v)}$ implies $(x,y) \in Ind^{\mathcal{S}}_{\{b\}}$. This together with $(x,y) \in Ind^{\mathcal{S}}_B$ gives $(x,y) \in Ind^{\mathcal{S}}_{B\cup\{b\}}$. Thus we obtain $y \in X$ as $x \in \underline{Ind^{\mathcal{S}}_{B\cup\{b\}}}(X)$.
(7) Follows from the fact that $Ind^{\mathcal{S}}_\emptyset = U \times U$. (8) is obvious. □

Remark 14.1. Let us explain the above proposition. 2 and 3 say that each object takes precisely one value for each attribute. According to 4, if an object is a positive element of a set X with respect to the indiscernibility relation corresponding to an attribute set C, then it remains so with respect to indiscernibility relations corresponding to any attribute set containing C. 5-6 relate the indiscernibility relations and attribute, attribute value pairs. According to 5, if an object w takes a value v for an attribute a, then every object indiscernible with w with respect to a, also takes the value v for the attribute a. 6 says that if an object w takes the value v for an attribute

b and every object indiscernible with w with respect to attributes from $B \cup \{b\}$ belongs to the set X, then every object indiscernible with w with respect to attributes from B, and which takes the value v for b also belongs to X.

We shall see later that these properties are actually characterizing properties of DIS-algebras. Thus, we propose the following notion of an abstract DIS-algebra.

Definition 14.6. An *abstract DIS-algebra of type* Ω is a tuple

$$\mathfrak{A} := (U, \wedge, \neg, 0, \{L_B\}_{B \subseteq \mathcal{A}}, \{d_\gamma\}_{\gamma \in \mathcal{D}}),$$

where $(U, \wedge, \neg, 0)$ is a Boolean algebra and L_B and d_γ are, respectively, unary and nullary (constant) operations on U satisfying the following:

(C_0) $L_B(x \wedge y) = L_B(x) \wedge L_B(y)$;
(C_1) $\bigvee_{v \in \mathcal{V}_a} d_{(a,v)} = 1$;
(C_2) $d_{(a,v)} \wedge d_{(a,u)} = 0$ when $v \neq u$;
(C_3) $L_C(x) \leq L_B(x)$ for $C \subseteq B \subseteq \mathcal{A}$;
(C_4) $d_{(a,v)} \leq L_{\{a\}}(d_{(a,v)})$;
(C_5) $d_{(b,v)} \wedge L_{B \cup \{b\}}(x) \leq L_B(d_{(b,v)} \rightarrow x)$;
(C_6) $L_\emptyset(x) \neq 0$ implies $x = 1$.
(C_7) $L_\emptyset(1) = 1$.

As a consequence of Proposition 14.1, the DIS-algebra $\mathcal{S}^*$ generated by a DIS $\mathcal{S}$ is an abstract DIS-algebra.

Let U_B be the dual of the operator L_B, that is, $U_B(x) := \neg L_B(\neg x)$. The following proposition presents a few properties of abstract DIS-algebras.

Proposition 14.2. *1.* $L_B(x) \leq x \leq U_B(x)$.
2. For $x \leq y$, $L_B(x) \leq L_B(y)$, *and* $U_B(x) \leq U_B(y)$.
3. $L_B(x) \vee L_B(y) \leq L_B(x \vee y)$, *and* $U_B(x) \wedge U_B(y) \geq U_B(x \wedge y)$.
4. $U_B(0) = 0$ *and* $L_B(1) = 1$.
5. $U_B(U_B(x)) = U_B(x)$ *and* $L_B(L_B(x)) = L_B(x)$.
6. $U_B(x \wedge U_B(y)) = U_B(x) \wedge U_B(y)$ *and* $L_B(x \vee L_B(y)) = L_B(x) \vee L_B(y)$.
7. $x \neq 0$ *implies* $U_\emptyset x = 1$.
8. $U_B(x \vee y) = U_B(x) \vee U_B(y)$.

Proof. We only provide the proof of items (1)-(5) for L_B.
(1) Using (C_6), we obtain the result for $B = \emptyset$. Next, we prove the result for singleton B. Let $B = \{a\}$ and consider an arbitrary $v \in \mathcal{V}_a$. Then, using (C_5), and the fact that $L_\emptyset(x) \leq x$, we obtain $d_{(a,v)} \wedge L_{\{a\}}(x) \leq d_{(a,v)} \rightarrow x$, and hence, $d_{(a,v)} \wedge L_{\{a\}}(x) \leq x$. Since this is true for all $v \in \mathcal{V}_a$, we obtain $\bigvee_{v \in \mathcal{V}_a} d_{(a,v)} \wedge L_{\{a\}}(x) \leq x$. Thus, (C_1) gives $L_{\{a\}}(x) \leq x$.
Now, assuming $L_B(x) \leq x$ and following exactly the above steps, one can prove $L_{B \cup \{b\}}(x) \leq x$.

(2) Let $x \leq y$. Then, $x \wedge y = x$, and hence, using (C_0), we obtain $L_B(x) \wedge L_B(y) = L_B(x)$. Therefore, $L_B(x) \leq L_B(y)$.

(3) Since $x \leq x \vee y$, using item (2), we obtain $L_B(x) \leq L_B(x \vee y)$. Similarly, we have $L_B(y) \leq L_B(x \vee y)$, and hence, we get $L_B(x) \vee L_B(y) \leq L_B(x \vee y)$.

(4) Follows from (C_3), and (C_7).

(5) First we note that for $x = 1$, $L_\emptyset(L_\emptyset(x)) = L_\emptyset(x) = 1$ (using (C_7)), and for $x \neq 1$, $L_\emptyset(L_\emptyset(x)) = L_\emptyset(x) = 0$ (using (C_6)). Next, we prove the result for singleton B, say $B = \{a\}$. Since $L_{\{a\}}(L_{\{a\}}(x)) \leq L_{\{a\}}(x)$ (by item 1), we just need to show the reverse inequality, that is, $L_{\{a\}}(x) \leq L_{\{a\}}(L_{\{a\}}(x))$. Here,

$$\begin{aligned} d_{(a,v)} \wedge L_{\{a\}}(x) &\leq L_\emptyset(\neg d_{(a,v)} \vee x) \quad \text{(by } (C_5)\text{)} \\ &= L_\emptyset L_\emptyset(\neg d_{(a,v)} \vee x) \\ &\leq L_{\{a\}} L_{\{a\}}(\neg d_{(a,v)} \vee x) \quad \text{(by } (C_3) \text{ and item 2)}. \end{aligned} \tag{14.1}$$

From (C_4) and item (2), we have $L_{\{a\}}(d_{(a,v)}) \leq L_{\{a\}} L_{\{a\}}(d_{(a,v)})$. Therefore, again using (C_4) we obtain

$$d_{(a,v)} \leq L_{\{a\}} L_{\{a\}}(d_{(a,v)}). \tag{14.2}$$

Thus, we have

$$\begin{aligned} d_{(a,v)} \wedge L_{\{a\}}(x) &\leq L_{\{a\}} L_{\{a\}}(\neg d_{(a,v)} \vee x) \wedge L_{\{a\}} L_{\{a\}}(d_{(a,v)}) \quad \text{(combining (14.1) and (14.2))} \\ &= L_{\{a\}} L_{\{a\}}((\neg d_{(a,v)} \vee x) \wedge d_{(a,v)}) \quad \text{(by } (C_0)\text{)} \\ &= L_{\{a\}} L_{\{a\}}(d_{(a,v)} \wedge x). \end{aligned} \tag{14.3}$$

Since (14.3) holds for all $v \in \mathcal{V}_a$, using item (3), we obtain

$$\bigvee_{v \in \mathcal{V}_a} d_{(a,v)} \wedge L_{\{a\}}(x) \leq L_{\{a\}} L_{\{a\}}\Big(\bigvee_{v \in \mathcal{V}_a} d_{(a,v)} \wedge x\Big). \tag{14.4}$$

Therefore, (C_1) gives $L_{\{a\}}(x) \leq L_{\{a\}}(L_{\{a\}}(x))$.
Next, assuming $L_B(x) \leq L_B L_B(x)$ and following exactly the above steps, we can prove $L_{B\cup\{a\}}(x) \leq L_{B\cup\{a\}}(L_{B\cup\{a\}}(x))$. This completes the proof. □

From Proposition 14.2, it is clear that U_B and L_B are, respectively, *closure* and *interior operators*. Moreover, the reduct $\mathfrak{A} := (U, \wedge, \neg, 0, \{L_B\}_{B\subseteq\mathcal{A}})$ is a *topological Boolean algebra* [4]. Furthermore, $(U, \wedge, \neg, 0, \{U_B\}_{B\subseteq\mathcal{A}})$ satisfies all the conditions of an abstract knowledge approximation algebra [8], except that in the latter case, the reduct $(U, \wedge, \neg, 0)$ is taken to be a complete atomic Boolean algebra, while we do not have that requirement.

Let us recall that a *cylindric algebra of dimension* $|\mathcal{A}|$ [13] is a structure

$$\mathfrak{A} := (U, \wedge, \neg, 0, \{\Lambda_a\}_{a\in\mathcal{A}}, \{\mu_{(a,b)}\}_{(a,b)\in\mathcal{A}\times\mathcal{A}}),$$

where $(U, \wedge, \neg, 0)$ is a Boolean algebra, and Λ_a, $\mu_{(a,b)}$ are, respectively, unary and nullary operations on U, such that

(L_1) $\Lambda_a(0) = 0$,
(L_2) $x \leq \Lambda_a(x)$,
(L_3) $\Lambda_a(x \wedge \Lambda_a(y)) = \Lambda_a(x) \wedge \Lambda_a(y)$,

(L_4) $\Lambda_a(\Lambda_b(x)) = \Lambda_b(\Lambda_a(x))$,
(L_5) $\mu_{(a,a)} = 1$,
(L_6) If $a \neq b, c$, then $\mu_{(b,c)} = \Lambda_a(\mu_{(b,a)} \wedge \mu_{(a,c)})$,
(L_7) If $a \neq b$, then $\Lambda_a(\mu_{(a,b)} \wedge x) \wedge \Lambda_a(\mu_{(a,b)} \wedge \neg x) = 0$.

The difference between the signature of an abstract DIS-algebra of type $(\mathcal{A}, \mathcal{V})$ and that of a cylindric algebra of dimension $|\mathcal{A}|$ is now clear. The cylindric algebra has unary and nullary operations corresponding to each element of $\mathcal{A}$, and $\mathcal{A} \times \mathcal{A}$, respectively. Whereas, in the case of abstract DIS-algebra, unary and nullary operations are indexed, respectively, over the sets $\wp(\mathcal{A})$ and $\mathcal{A} \times \mathcal{V}$. Moreover, operators U_B of an abstract DIS-algebra satisfy (L_1)–(L_3), but may fail to satisfy (L_4). (L_5)–(L_7) do not make sense in the case of abstract DIS-algebras. However, the BAO $(U, \wedge, \neg, 0, U_B)$ obtained from an abstract DIS-algebra is a cylindric algebra of dimension 1.

14.3 Representation Theorem for Abstract DIS-Algebras

The proof of the representation theorem for abstract knowledge approximation algebras given in [8] makes use of the completeness and atomicity properties of the Boolean reduct of the algebra. In fact, the embedding of an abstract knowledge approximation algebra $\mathfrak{A}$ is given in an extension of the power set algebra over the set $At(\mathfrak{A})$ of atoms of $\mathfrak{A}$. But in the case of abstract DIS-algebras, the Boolean reduct may not be complete and atomic, and hence, this technique will not work. We use *prime filters* [7] for our purpose.

Recall that a *filter* of a Boolean algebra $\mathfrak{A} := (U, \wedge, \sim, 0)$ is a subset F of U such that (i) $1 \in F$, (ii) if $a, b \in F$, then $a \wedge b \in F$, (iii) if $a \in F$ and $a \leq b$, then $b \in F$. A filter is *proper* if it does not contain the smallest element 0. A proper filter is *prime* if $a \vee b \in F$ implies that at least one of a and b belongs to F. We note that for a prime filter F, we have

- $a \rightarrow b, a \in F$ implies $b \in F$, and
- $a \rightarrow b \notin F$ implies $a \in F$ and $b \notin F$.

We shall require these facts later.

Let $PF(\mathfrak{A})$ denote the set of all prime filters of $\mathfrak{A}$.

Let us consider an abstract DIS-algebra $\mathfrak{A} := (U, \wedge, \neg, 0, \{L_B\}_{B \subseteq \mathcal{A}}, \{d_\alpha\}_{\alpha \in \mathcal{D}})$. $\mathfrak{A}$ determines a unique DIS $\mathfrak{A}_*$ as follows.
Consider the mapping $f_{\mathfrak{A}} : PF(\mathfrak{A}) \times \mathcal{A} \rightarrow \mathcal{V}$ such that

$$f_{\mathfrak{A}}(\Gamma, a) = v \text{ if and only if } d_{(a,v)} \in \Gamma.$$

Conditions (C_1) and (C_2) in Definition 14.6 guarantee that $f_{\mathfrak{A}}$ is a total function. Thus, we obtain the DIS $\mathfrak{A}_* := (PF(\mathfrak{A}), \mathcal{A}, \mathcal{V}, f_{\mathfrak{A}})$. $\mathfrak{A}_*$ determines the lower approximation operators $\underline{Ind_B^{\mathfrak{A}_*}}$, $B \subseteq \mathcal{A}$, on $\wp(PF(\mathfrak{A}))$.

We also recall that the reduct $(U, \wedge, \neg, 0, \{L_B\}_{B \subseteq \mathcal{A}})$ of an abstract DIS-algebra $\mathfrak{A} := (U, \wedge, \neg, 0, \{L_B\}_{B \subseteq \mathcal{A}}, \{d_\alpha\}_{\alpha \in \mathcal{D}})$ determines a *complex algebra* [7] as follows.

For each $B \subseteq \mathcal{A}$, let us consider the binary relation $Q_B^{\mathfrak{A}} \subseteq PF(\mathfrak{A}) \times PF(\mathfrak{A})$:

$$(\Gamma, \Delta) \in Q_B^{\mathfrak{A}} \text{ if and only if } L_B(x) \in \Gamma \text{ implies } x \in \Delta.$$

The relations $Q_B^{\mathfrak{A}}$ are used to define the operators $m_B^{\mathfrak{A}} : \wp(PF(\mathfrak{A})) \to \wp(PF(\mathfrak{A}))$:

$$m_B^{\mathfrak{A}}(X) := \{\Gamma \in PF(\mathfrak{A}) : \text{ for all } \Delta \text{ such that } (\Gamma, \Delta) \in Q_B^{\mathfrak{A}},\ \Delta \in X\}.$$

The complex algebra corresponding to the reduct $(U, \wedge, \neg, 0, \{L_B\}_{B \subseteq \mathcal{A}})$ of the abstract DIS-algebra $\mathfrak{A}$ is given by extending the power set algebra over $PF(\mathfrak{A})$ with the operators $m_B^{\mathfrak{A}}$.

So, an abstract DIS-algebra $\mathfrak{A}$, on the one hand, determines the lower approximation operators $\underline{Ind_B^{\mathfrak{A}*}}$. On the other hand, it gives rise to the complex algebra with operators $m_B^{\mathfrak{A}}$. Is there any relationship between the operators $m_B^{\mathfrak{A}}$, and the lower approximation operators $\underline{Ind_B^{\mathfrak{A}*}}$? In fact, we shall now show that for each $B \subseteq \mathcal{A}$, the operators $m_B^{\mathfrak{A}}$ and $\underline{Ind_B^{\mathfrak{A}*}}$ are the same. This result will also lead us to the desired representation theorem. Let us begin with the following proposition listing a few properties of the relations $Q_B^{\mathfrak{A}}$.

Proposition 14.3

1. $Q_B^{\mathfrak{A}} \subseteq Q_C^{\mathfrak{A}}$ *for* $C \subseteq B \subseteq \mathcal{A}$.
2. $d_{(b,v)} \in \Gamma \cap \Delta$ *for some* $v \in \mathcal{V}_b$ *if and only if* $(\Gamma, \Delta) \in Q_{\{b\}}^{\mathfrak{A}}$.
3. *If* $(\Gamma, \Delta) \in Q_B^{\mathfrak{A}}$ *and* $d_{(b,v)} \in \Gamma \cap \Delta$ *for some* $v \in \mathcal{V}_b$, *then* $(\Gamma, \Delta) \in Q_{B \cup \{b\}}^{\mathfrak{A}}$.
4. $Q_{\emptyset}^{\mathfrak{A}} = PF(\mathfrak{A}) \times PF(\mathfrak{A})$.
5. $Q_B^{\mathfrak{A}} = \bigcap_{b \in B} Q_{\{b\}}^{\mathfrak{A}}$.

Proof (1) is a direct consequence of (C_3). Let us prove (2). First suppose $d_{(b,v)} \in \Gamma \cap \Delta$ for some $v \in \mathcal{V}_b$, and let $L_{\{b\}}(x) \in \Gamma$. We need to show $x \in \Delta$. Using the properties of filters, we obtain $d_{(b,v)} \wedge L_{\{b\}}(x) \in \Gamma$, and hence, by (C_5) with $B = \emptyset$, we obtain $L_{\emptyset}(d_{(b,v)} \to x) \in \Gamma$. This shows that $L_{\emptyset}(d_{(b,v)} \to x) \neq 0$, and hence, by (C_6), we obtain $d_{(b,v)} \to x = 1$. Therefore, we have $d_{(b,v)} \to x \in \Delta$. Finally using the fact that $d_{(b,v)} \in \Delta$, we obtain $x \in \Delta$.

Conversely, suppose $(\Gamma, \Delta) \in Q_{\{b\}}^{\mathfrak{A}}$. By (C_1), there exists a $v \in \mathcal{V}_b$ such that $d_{(b,v)} \in \Gamma$. Therefore, using (C_4), we obtain $L_{\{b\}}(d_{(b,v)}) \in \Gamma$, and so $d_{(b,v)} \in \Delta$.

Let us now prove (3). Suppose $(\Gamma, \Delta) \in Q_B^{\mathfrak{A}}$ and $d_{(b,v)} \in \Gamma \cap \Delta$ for some $v \in \mathcal{V}_b$. Further, suppose $L_{B \cup \{b\}}(x) \in \Gamma$. We need to show $x \in \Delta$. Due to the given conditions, we obtain $d_{(b,v)} \wedge L_{B \cup \{b\}}(x) \in \Gamma$, and hence, by (C_5), $L_B(d_{(b,v)} \to x) \in \Gamma$. This gives $d_{(b,v)} \to x \in \Delta$, as $(\Gamma, \Delta) \in Q_B^{\mathfrak{A}}$. As $d_{(b,v)} \in \Delta$, $x \in \Delta$.

(4) is obvious due to (C_6). Let us now move to (5). From (1), we obtain $Q_B^{\mathfrak{A}} \subseteq \bigcap_{b \in B} Q_{\{b\}}^{\mathfrak{A}}$. It is also not difficult to see that the reverse inclusion holds when $|B| \leq 1$. To complete the proof, let us assume that the reverse inclusion holds for B, and prove it for $B \cup \{a\}$. Let $(\Gamma, \Delta) \in \bigcap_{b \in B \cup \{a\}} Q_{\{b\}}^{\mathfrak{A}}$. We need to show $(\Gamma, \Delta) \in Q_{B \cup \{a\}}^{\mathfrak{A}}$. Using

(2) and the fact that $(\Gamma,\Delta)\in Q^{\mathfrak{A}}_{\{a\}}$, we obtain $d_{(a,v)}\in\Gamma\cap\Delta$ for some v. Now (3) gives $(\Gamma,\Delta)\in Q^{\mathfrak{A}}_{B\cup\{a\}}$. □

Theorem 14.1. *Let* $\mathfrak{A}:=(U,\wedge,\neg,0,\{L_B\}_{B\subseteq\mathcal{A}},\{d_\gamma\}_{\gamma\in\mathcal{D}})$ *be an abstract DIS-algebra. Then, the following hold for each* $B\subseteq\mathcal{A}$.

1. $Ind^{\mathfrak{A}_*}_B = Q^{\mathfrak{A}}_B$.
2. $Ind^{\mathfrak{A}_*}_B = m^{\mathfrak{A}}_B$.

Proof. (2) is a direct consequence of (1). So, we only prove (1). Due to (5) of Proposition 14.3, it is enough to prove (1) for singleton B. Let $(\Gamma,\Delta)\in Ind^{\mathfrak{A}_*}_{\{b\}}$. This implies $f_{\mathfrak{A}}(\Gamma,b)=f_{\mathfrak{A}}(\Delta,b)=v$ for some v. Therefore, from the definition of $f_{\mathfrak{A}}$, we obtain $d_{(b,v)}\in\Gamma\cap\Delta$. Now using (2) of Proposition 14.3, we obtain $(\Gamma,\Delta)\in Q^{\mathfrak{A}}_{\{b\}}$ as desired. The reverse inclusion can be proved similarly. □

Theorem 14.2 (Representation theorem for abstract DIS-algebras). *Let* $\mathfrak{A}:=(U,\wedge,\neg,0,\{L_B\}_{B\subseteq\mathcal{A}},\{d_\gamma\}_{\gamma\in\mathcal{D}})$ *be an abstract DIS-algebra. Then the mapping* $\Psi:U\to\wp(PF(\mathfrak{A}))$ *given by*

$$\Psi(x):=\{\Gamma\in PF(\mathfrak{A}):x\in\Gamma\},\ x\in U,$$

is an embedding of $\mathfrak{A}$ *into* $(\mathfrak{A}_*)^*$.

Proof. It is not difficult to see that $\Psi(d_\gamma)=c^{\mathfrak{A}_*}_\gamma$, $\gamma\in\mathcal{D}$. Due to item (2) of Theorem 14.1, the rest follows in the lines of the proof of Jónnson-Tarski theorem (cf. [7]). □

14.4 Logics for Deterministic Information Systems

Let us consider a language $\mathcal{L}$ consisting of a countable set $Var:=\{p,q,r,\ldots\}$ of variables, a binary operator $\wedge$, unary operators $\neg$, $\mathbf{L}_B$ and constants 0, $\mathbf{d}_{(a,v)}$, where $B\subseteq\mathcal{A}$, $(a,v)\in\mathcal{D}$. The *well-formed formulae* (wffs) of $\mathcal{L}$ are defined recursively:

$$\alpha:=p\in Var\mid 0\mid\mathbf{d}_{(a,v)}\mid\neg\alpha\mid\alpha\wedge\beta\mid\mathbf{L}_B\alpha.$$

Now consider an abstract DIS-algebra $\mathfrak{A}:=(U,\wedge,\neg,0,\{L_B\}_{B\subseteq\mathcal{A}},\{d_\gamma\}_{\gamma\in\mathcal{D}})$. An *assignment* for $\mathfrak{A}$ is a map $V:Var\to U$. V can be extended to a mapping $\tilde{V}$ from the set of all $\mathcal{L}$-wffs to U in the obvious way: $0,d_{(a,v)},L_B$ correspond, respectively, to $0,d_{(a,v)},L_B$. An *equation* $\alpha\approx\beta$ is said to hold in $\mathfrak{A}$, denoted as $\mathfrak{A}\models\alpha\approx\beta$, if $\tilde{V}(\alpha)=\tilde{V}(\beta)$ for all V.

The notion of equivalence defined above can be used to realize laws related to DISs and approximations. For instance, one may easily verify that, for all $\mathfrak{A}$,

$$\mathfrak{A}\models((d_{(b,v)}\wedge L_{B\cup\{b\}}(p))\to(L_B(d_{(b,v)}\to p)))\approx 1,$$

where $\to$ is the logical connective for implication defined in the usual way: $\alpha\to\beta:=\neg\alpha\vee\beta$. The representation theorem also leads to the complete axiomatization for the semantic notion of equivalence in DIS-algebras generated by DISs. More formally

speaking, using Birkhoff's completeness theorem for equational logic [6], one can prove that if $\alpha \approx \beta$ holds in all DIS-algebras generated by DISs, then $\alpha \approx \beta$ is derivable from the equations (C_0)–(C_7).

We end this section with the remark that abstract DIS-algebras are actually an algebraic counterpart of the logic LIS of deterministic information systems proposed in [15]. In order to see it, recall that LIS-wffs are given by the scheme

$$(a,v) \in \mathcal{D} \mid p \in Var \mid \neg\alpha \mid \alpha \wedge \beta \mid [I(B)]\alpha.$$

Clearly, LIS-wffs are identifiable with the $\mathcal{L}$-wffs by the bijection θ mapping p to p, $\gamma \in \mathcal{D}$ to d_γ and $[I(B)]\alpha$ to $L_B(\theta(\alpha))$. Using Theorem 14.2, one can then show that $\vdash_{LIS} \alpha$ if and only if $\theta(\alpha) \approx 1$ holds in all abstract DIS-algebras.

14.5 Algebra for Incomplete Information Systems

We recall the definitions (Definition 14.2) of information systems and incomplete information systems. As mentioned in Section 14.1, in the case of (incomplete) information systems, a similarity relation is used as the distinguishability relation, rather than the indiscernibility relation. As in the case of the indiscernibility relation, the similarity relation $Sim_B^{\mathcal{S}}$ determines the unary operation $\underline{Sim_B^{\mathcal{S}}}$ on $\wp(U)$ mapping a set X to $\underline{X}_{Sim_B^{\mathcal{S}}}$. Thus, we extend the notion of DIS-algebra to incorporate the similarity relation and define *IS-algebra of type* Ω *generated by the information system* $\mathcal{S}$ as the structure

$$\mathcal{S}^* := (\wp(U), \cap, \sim, \emptyset, \{\underline{Ind_B^{\mathcal{S}}}\}_{B\subseteq\mathcal{A}}, \{\underline{Sim_B^{\mathcal{S}}}\}_{B\subseteq\mathcal{A}}, \{c_\gamma^{\mathcal{S}}\}_{\gamma\in\mathcal{D}}).$$

Moreover, an *abstract IS-algebra* of type Ω is a tuple

$$\mathfrak{A} := (U, \wedge, \neg, 0, \{L_B\}_{B\subseteq\mathcal{A}}, \{S_B\}_{B\subseteq\mathcal{A}}, \{d_\gamma\}_{\gamma\in\mathcal{D}}),$$

where $(U, \wedge, \neg, 0, \{L_B\}_{B\subseteq\mathcal{A}}, \{d_\gamma\}_{\gamma\in\mathcal{D}})$ is an abstract DIS-algebra and $\mathcal{S}_B, B \subseteq \mathcal{A}$ are unary operations on U such that

(C_8) $S_C(x) \leq S_B(x)$ for $C \subseteq B \subseteq \mathcal{A}$,
(C_9) $d_{(a,v)} \leq S_{\{a\}}(d_{(a,v)} \vee d_{(a,*)})$, where $v \neq *$,
(C_{10}) $d_{(b,v)} \wedge S_{B\cup\{b\}}(x) \leq S_B((d_{(b,v)} \vee d_{(b,*)}) \rightarrow x)$, where $v \neq *$,
(C_{11}) $d_{(b,*)} \wedge S_{B\cup\{b\}}(x) \leq S_B(x)$.

If the abstract IS-algebra satisfies the additional condition

(C_{12}) $\bigvee_{\alpha\in\mathcal{D}'} d_\gamma \neq 1$, where $\mathcal{D}' := \{(a,v) : a \in \mathcal{A}, v \in \mathcal{V}_a \setminus \{*\}\}$,

then it will be called an *abstract IIS-algebra*.

Similarly to Theorem 14.2, we obtain the following representation theorem.

Theorem 14.3 (Representation theorem for abstract (I)IS-algebras)
Every abstract IS-algebra (IIS-algebra) is isomorphic to a subalgebra of $\mathcal{S}^*$ corresponding to some IS (IIS) S.

Using the above representation theorem, as in the case of DIS, one can show that the abstract IS-algebra is an algebraic counterpart of the logic proposed in [16] for information systems. Moreover, the theorem also gives an equational logic for ISs consisting of the equations $(C_0)-(C_{11})$.

14.6 Algebra for Non-deterministic Information Systems

In this section, our aim is to propose an algebraic formalism for NISs that would also capture the notion of set approximations with respect to different relations defined on NISs. For the moment, we restrict ourselves to indiscernibility, similarity, and inclusion relations (cf. Section 14.1). In Section 14.5, we have seen that the results obtained in Sections 14.2-14.4 for DISs can be extended in a natural way to obtain an algebra for ISs. The situation is not so simple for NISs. In fact, we need axioms which relate approximations relative to different sets of attributes and attribute-value pairs. In the case of DISs, axioms $(C_3)-(C_5)$ serve the purpose, but axioms (C_4), (C_5) are not sound (as will be illustrated in Example 14.1) when we move to NISs. Therefore, we need replacements for these axioms and for this purpose, we shall take the help of unary operators which provide *names* to objects. These operators will help to reason about the equality of objects. Thus, we have the following notion of an algebra generated from a NIS.

As before, let us consider finite sets $\mathcal{A}$ and $\mathcal{V}_a$, $a \in \mathcal{A}$, of attributes and attribute values and let Ω be the tuple $(\mathcal{A}, \bigcup_{a\in\mathcal{A}} \mathcal{V}_a)$. Recall that the attribute-value pairs $(a,v) \in \mathcal{D}$ represent the collection of objects taking the value v for the attribute a. As objects take *a set of* attribute values in the case of NISs, we consider the set $\mathcal{D}^a := \{(a,V) : V \subseteq \mathcal{V}_a\}$, for each $a \in \mathcal{A}$. Thus, each element (a,v) in $\mathcal{D}$ may be viewed as the element $(a,\{v\})$ of $\mathcal{D}^a$. Observe that $|\prod_{a\in\mathcal{A}} \mathcal{D}^a|$ is finite. Moreover, for any NIS $\mathcal{S} := (U,\mathcal{A},\mathcal{V},f)$, $|U/Ind^{\mathcal{S}}_{\mathcal{A}}| \leq |\prod_{a\in\mathcal{A}} \mathcal{D}^a|$. Let $\Theta := \{i,j,\ldots\}$ be a finite set of "nominals" with $|\Theta| = |\prod_{a\in\mathcal{A}} \mathcal{D}^a|$.

Definition 14.7. Let $\mathcal{S} := (U,\mathcal{A},\mathcal{V},f)$ be a non-deterministic information system. A *non-deterministic information system algebra (in brief,* NIS-algebra*) of type* Ω *generated by the non-deterministic information system* $\mathcal{S}$ is a structure

$$\mathcal{S}^* := (\wp(U), \cap, \sim, \emptyset, \{\underline{Ind^{\mathcal{S}}_B}\}_{B\subseteq\mathcal{A}}, \{\underline{Sim^{\mathcal{S}}_B}\}_{B\subseteq\mathcal{A}}, \{\underline{In^{\mathcal{S}}_B}\}_{B\subseteq\mathcal{A}}, \{c^{\mathcal{S}}_\gamma\}_{\gamma\in\mathcal{D}}, \{c^{\mathcal{S}}_i\}_{i\in\Theta}),$$

where $\underline{Ind^{\mathcal{S}}_B}$, $\underline{Sim^{\mathcal{S}}_B}$, $\underline{In^{\mathcal{S}}_B}$ are operators on $\wp(U)$ mapping a set X to $\underline{X}_{Ind^{\mathcal{S}}_B}$, $\underline{X}_{Sim^{\mathcal{S}}_B}$, and $\underline{X}_{In^{\mathcal{S}}_B}$ respectively, for $\gamma := (a,v) \in \mathcal{D}$, $c^{\mathcal{S}}_\gamma$ is the nullary operation (constant) given by the subset $\{x \in U : f(x,a) = v\}$ of U, and $c^{\mathcal{S}}_i$ are nullary operations on $\wp(U)$ satisfying the following.

(N1) $U/Ind^{\mathcal{S}}_{\mathcal{A}} \subseteq \{c^{\mathcal{S}}_i : i \in \Theta\}$.
(N2) $c^{\mathcal{S}}_i \cap c^{\mathcal{S}}_j = \emptyset$, for $i \neq j$.
(N3) $c^{\mathcal{S}}_i \in U/Ind^{\mathcal{S}}_{\mathcal{A}} \cup \{\emptyset\}$.
(N4) If $(x,y) \notin Ind^{\mathcal{S}}_{\mathcal{A}}$, $[x]_{Ind^{\mathcal{S}}_{\mathcal{A}}} = c^{\mathcal{S}}_i$ and $[y]_{Ind^{\mathcal{S}}_{\mathcal{A}}} = c^{\mathcal{S}}_j$, then $i \neq j$.

From conditions (N1)–(N4), it is clear that the nullary operators $c^{\mathcal{S}}_i$ are used to name the equivalence classes of $Ind^{\mathcal{S}}_{\mathcal{A}}$ such that different equivalence classes are provided with different names. This could also be viewed as providing names to elements of the set U such that elements belonging to the same equivalence class of the relation $Ind^{\mathcal{S}}_{\mathcal{A}}$ are provided with the same name and elements belonging to different equivalence classes of $Ind^{\mathcal{S}}_{\mathcal{A}}$ have different names. Observe that due to size of the set Θ, we have enough nominals to achieve this task. Also note that the reduct $(\wp(U), \cap, \sim, \emptyset, \{\underline{Ind^{\mathcal{S}}_B}\}_{B\subseteq\mathcal{A}}, \{\underline{Sim^{\mathcal{S}}_B}\}_{B\subseteq\mathcal{A}}, \{\underline{In^{\mathcal{S}}_B}\}_{B\subseteq\mathcal{A}}, \{c^{\mathcal{S}}_\gamma\}_{\gamma\in\mathcal{D}})$ of $\mathcal{S}^*$ is determined uniquely, and thus, a NIS $\mathcal{S}$ can generate two distinct NIS-algebras which can differ only with respect to naming of the objects, that is, with respect to nullary operators corresponding to the elements from the set Θ.

We would like to mention here that the above idea of naming objects (elements of the domain) is not new. In fact, it is the main idea of hybrid logics (cf. [7]). The idea of naming objects is also used by Konikowska [18, 19] in the proposals of modal logics for information systems and rough set theory. The main difference between these and our way of naming is that we are providing the same name to the elements belonging to the same equivalence class of $Ind^{\mathcal{S}}_{\mathcal{A}}$ and different names to objects belonging to different classes. In other words, as mentioned above, we are effectively providing names to the equivalence classes instead of individual elements.

The following proposition lists a few properties of NIS-algebras.

Proposition 14.4. *Let X be a subset of the domain U.*

1. $L_C(X) \subseteq L_B(X)$ *for* $C \subseteq B \subseteq \mathcal{A}$, $L \in \{Ind^{\mathcal{S}}, Sim^{\mathcal{S}}, In^{\mathcal{S}}\}$.
2. $c^{\mathcal{S}}_{(a,v)} \subseteq L_{\{a\}}(c^{\mathcal{S}}_{(a,v)})$, $L \in \{Ind^{\mathcal{S}}, In^{\mathcal{S}}\}$.
3. $\sim c^{\mathcal{S}}_{(a,v)} \subseteq Ind^{\mathcal{S}}_{\{a\}}(\sim c^{\mathcal{S}}_{(a,v)})$.
4. $c^{\mathcal{S}}_i \subseteq Sim^{\mathcal{S}}_{\{a\}}(\bigcup_{v\in\mathcal{V}_a}(c^{\mathcal{S}}_{(a,v)} \cap Ind^{\mathcal{S}}_{\emptyset}(c^{\mathcal{S}}_i \to c^{\mathcal{S}}_{(a,v)})))$.
5. $c^{\mathcal{S}}_i \cap Ind^{\mathcal{S}}_{B\cup\{b\}}(X) \subseteq Ind^{\mathcal{S}}_B\left(\bigcap_{v\in\mathcal{V}_b}\left(c^{\mathcal{S}}_{(b,v)} \leftrightarrow Ind^{\mathcal{S}}_{\emptyset}(c^{\mathcal{S}}_i \to c^{\mathcal{S}}_{(b,v)})\right) \to X\right)$.
6. $c^{\mathcal{S}}_{(b,v)} \cap Sim^{\mathcal{S}}_{B\cup\{b\}}(X) \subseteq Sim^{\mathcal{S}}_B(c^{\mathcal{S}}_{(b,v)} \to X)$.
7. $c^{\mathcal{S}}_i \cap In^{\mathcal{S}}_{B\cup\{b\}}(X) \subseteq In^{\mathcal{S}}_B\left(\bigcap_{v\in\mathcal{V}_b}\left(Ind^{\mathcal{S}}_{\emptyset}(c^{\mathcal{S}}_i \to c^{\mathcal{S}}_{(b,v)}) \to c^{\mathcal{S}}_{(b,v)}\right) \to X\right)$.
8. $Ind^{\mathcal{S}}_{\emptyset}(X) = Sim^{\mathcal{S}}_{\emptyset}(X) = In^{\mathcal{S}}_{\emptyset}(X)$.
9. $Ind^{\mathcal{S}}_{\emptyset}(X) \neq \emptyset$ *implies* $X = U$.
10. $c^{\mathcal{S}}_i \cap c^{\mathcal{S}}_{(a,v)} \subseteq Ind^{\mathcal{S}}_{\emptyset}(c^{\mathcal{S}}_i \to c^{\mathcal{S}}_{(a,v)})$.
11. $c^{\mathcal{S}}_i \cap \sim c^{\mathcal{S}}_{(a,v)} \subseteq Ind^{\mathcal{S}}_{\emptyset}(c^{\mathcal{S}}_i \to \sim c^{\mathcal{S}}_{(a,v)})$.
12. $\bigcup_{i\in\Theta} c^{\mathcal{S}}_i = U$.
13. $\sim c^{\mathcal{S}}_i \cup \sim c^{\mathcal{S}}_j = U$ *for* $i \neq j$.

14. $c_i^{\mathcal{S}} \subseteq Ind_{\mathcal{A}}^{\mathcal{S}} c_i^{\mathcal{S}}$.
15. $Ind_{\emptyset}^{\mathcal{S}}(U) = U$.
16. $L_B(X \cap Y) = L_B(X) \cap L_B(Y)$ *for* $L \in \{Ind^{\mathcal{S}}, Sim^{\mathcal{S}}, In^{\mathcal{S}}\}$.

Proof. We only provide the proof of items (4) and (5).
(4) Let $x \in c_i^{\mathcal{S}}$. We need to show $x \in Sim_{\{a\}}^{\mathcal{S}}(\bigcup_{v \in \mathcal{V}_a}(c_{(a,v)}^{\mathcal{S}} \cap Ind_{\emptyset}^{\mathcal{S}}(c_i^{\mathcal{S}} \to c_{(a,v)}^{\mathcal{S}})))$. Let y be such that $(x,y) \in Sim_{\{a\}}^{\mathcal{S}}$, that is, there exists a $v \in f(x,a) \cap f(y,a)$. In order to prove the result, it is enough to show that $y \in c_{(a,v)}^{\mathcal{S}} \cap Ind_{\emptyset}^{\mathcal{S}}(c_i^{\mathcal{S}} \to c_{(a,v)}^{\mathcal{S}})$. Since $v \in f(y,a)$, we obtain $y \in c_{(a,v)}^{\mathcal{S}}$. In order to prove $y \in Ind_{\emptyset}^{\mathcal{S}}(c_i^{\mathcal{S}} \to c_{(a,v)}^{\mathcal{S}})$, let us take an arbitrary $z \in c_i^{\mathcal{S}}$, and we prove $z \in c_{(a,v)}^{\mathcal{S}}$. Since $x, z \in c_i^{\mathcal{S}}$, by the condition (N3) of Definition 14.7, we obtain $z \in [x]_{Ind_{\mathcal{A}}^{\mathcal{S}}}$. Since $v \in f(x,a)$, we obtain $v \in f(z,a)$, and hence $z \in c_{(a,v)}^{\mathcal{S}}$.

(5) Let $x \in c_i^{\mathcal{S}} \cap Ind_{B \cup \{b\}}^{\mathcal{S}}(X)$. Thus, by the condition (N3) of Definition 14.7, we obtain

$$c_i^{\mathcal{S}} = [x]_{Ind_{\mathcal{A}}^{\mathcal{S}}}. \tag{14.5}$$

Let us consider an arbitrary y such that $(x,y) \in Ind_B^{\mathcal{S}}$, and

$$y \in \bigcap_{v \in \mathcal{V}_b} \left(c_{(b,v)}^{\mathcal{S}} \leftrightarrow Ind_{\emptyset}^{\mathcal{S}}(c_i^{\mathcal{S}} \to c_{(b,v)}^{\mathcal{S}}) \right). \tag{14.6}$$

We need to show $y \in X$. Since $x \in Ind_{B \cup \{b\}}^{\mathcal{S}}(X)$, it is enough to show that $(x,y) \in Ind_{\{b\}}^{\mathcal{S}}$, that is, $v \in f(x,b)$ if and only if $v \in f(y,b)$.
First suppose $v \in f(x,b)$. Then we have $x \in c_{(b,v)}^{\mathcal{S}}$. Therefore, from (14.5), we obtain $c_i^{\mathcal{S}} = [x]_{Ind_{\mathcal{A}}^{\mathcal{S}}} \subseteq [x]_{Ind_{\{b\}}^{\mathcal{S}}} \subseteq c_{(b,v)}^{\mathcal{S}}$, and hence $Ind_{\emptyset}^{\mathcal{S}}(c_i^{\mathcal{S}} \to c_{(b,v)}^{\mathcal{S}}) = U$. Therefore, (14.6) gives $y \in c_{(b,v)}^{\mathcal{S}}$, that is, $v \in f(y,b)$.
Now suppose $v \in f(y,b)$, that is, $y \in c_{(b,v)}^{\mathcal{S}}$. Then from (14.6), we obtain $y \in Ind_{\emptyset}^{\mathcal{S}}(c_i^{\mathcal{S}} \to c_{(b,v)}^{\mathcal{S}})$. This implies that for all $z \in c_i^{\mathcal{S}}$, we have $z \in c_{(b,v)}^{\mathcal{S}}$. Since $x \in c_i^{\mathcal{S}}$, we obtain $x \in c_{(b,v)}^{\mathcal{S}}$. □

Remark 14.2. Note that 8, 9 and 15 list the properties of lower approximations with respect to indiscernibility, similarity and inclusion relations relative to the empty set of attributes. The lower approximation of a proper subset of the domain U with respect to any of these relations relative to the empty set of attributes is empty and that of the domain U is U itself. 10–14 give the rules followed in naming the objects. According to 10, 11, objects with the same name take the same values for each attribute. Properties 12, 13 guarantee that each object is assigned precisely one name. According to 14, objects belonging to the same equivalence class with respect to the indiscernibility relation relative to $\mathcal{A}$ have the same names. Apart from these, there

are properties which relate approximations relative to different sets of attributes and attribute-value pairs. We have 1–3 and 5 serving this purpose for the indiscernibility relation. For the similarity relation, we have 1, 4 and 6. 1, 2 and 7 serve the purpose for the inclusion relation.

2 is similar to 4 of Proposition 14.1. According to it, if an object w takes the value v for an attribute a, then every object indiscernible to w relative to a, also takes the value v for a. This is also true for the inclusion relation, but not true for the similarity relation, as will be illustrated by Example 14.1 below. In the case of indiscernibility, we have more: if an object w does not take the value v for an attribute a, then every object indiscernible to w relative to a, also does not take the value v for a. This is precisely what 3 says. 3 is also not true for the similarity relation, and we have 4 for it. According to 4, if an object w is named i, then for every object w' similar to w relative to an attribute a, there exists an attribute value v such that w' and every object named i take the value v for a. Thus, it means there exists an attribute value v such that w and w' take this value for a. 6 is similar to 5 of Proposition 14.1, which is explained in Remark 14.1. The fact captured by 6 is not true for indiscernibility and inclusion relations defined on NISs – this is illustrated by Example 14.1 below. Instead, we have 5 and 7 for these relations. According to 5, if an object w is named i and every object indiscernible with w relative to the attribute set $B \cup \{b\}$ belongs to X, then every object w' such that (i) w' is indiscernible with w relative to the attribute set B and (ii) w' takes precisely those values for b which are taken by the objects named i for b, also belongs to X. The interpretation of 7 for inclusion relation is very similar to the above interpretation of 5 except that in (ii) we have a weaker condition. It says that if an object w is named i and for every w^0 with $(w, w^0) \in In^S_{B\cup\{b\}}$, we have $w^0 \in X$, then, every object w' such that (i) $(w, w') \in In^S_B$ and (ii) w' takes a value v for the attribute b whenever an object named i does so, also belongs to X.

We shall find later that properties 1-16 given in Proposition 14.4 are actually characterizing properties of NIS-algebras. Thus, we propose the following abstract algebra for NISs.

Definition 14.8. An *abstract NIS-algebra of type* Ω is a tuple

$$\mathfrak{A} := (U, \wedge, \neg, 0, \{I_B\}_{B\subseteq\mathcal{A}}, \{S_B\}_{B\subseteq\mathcal{A}}, \{N_B\}_{B\subseteq\mathcal{A}}, \{d_\gamma\}_{\gamma\in\mathcal{D}}, \{d_i\}_{i\in\Theta}),$$

where $(U, \wedge, \neg, 0)$ is a Boolean algebra, I_B, S_B, N_B are unary operations, and d_γ, d_i are nullary (constant) operations on U satisfying the following.

(N_1) $L_C(x) \leq L_B(x)$ for $C \subseteq B \subseteq \mathcal{A}$, $L \in \{I, S, N\}$.
(N_2) $d_{(a,v)} \leq L_{\{a\}}(d_{(a,v)})$, $L \in \{I, N\}$.
(N_3) $\neg d_{(a,v)} \leq I_{\{a\}}(\neg d_{(a,v)})$.
(N_4) $d_i \leq S_{\{a\}}(\bigvee_{v\in\mathcal{V}_a}(d_{(a,v)} \wedge I_\emptyset(d_i \to d_{(a,v)})))$.
(N_5) $d_i \wedge I_{B\cup\{b\}}(x) \leq I_B\left(\bigwedge_{v\in\mathcal{V}_b}\left(d_{(b,v)} \leftrightarrow I_\emptyset(d_i \to d_{(b,v)})\right) \to x\right)$.
(N_6) $d_{(b,v)} \wedge S_{B\cup\{b\}}(x) \leq S_B(d_{(b,v)} \to x)$.
(N_7) $d_i \wedge N_{B\cup\{b\}}(x) \leq N_B\left(\bigwedge_{v\in\mathcal{V}_b}\left(I_\emptyset(d_i \to d_{(b,v)}) \to d_{(b,v)}\right) \to x\right)$.

(N_8) $I_\emptyset(x) = S_\emptyset(x) = N_\emptyset(x)$.
(N_9) $I_\emptyset(x) \neq 0$ implies $x = 1$.
(N_{10}) $d_i \wedge d_{(a,v)} \leq I_\emptyset(d_i \rightarrow d_{(a,v)})$.
(N_{11}) $d_i \wedge \neg d_{(a,v)} \leq I_\emptyset(d_i \rightarrow \neg d_{(a,v)})$.
(N_{12}) $\bigvee_{i\in\Theta} d_i = 1$.
(N_{13}) $\neg d_i \vee \neg d_j = 1$ for $i \neq j$.
(N_{14}) $d_i \leq I_{\mathcal{A}} d_i$.
(N_{15}) $I_\emptyset(1) = 1$.
(N_{16}) $L_B(x \wedge y) = L_B(x) \wedge L_B(y)$ for $L \in \{I, S, N\}$.

We note that a NIS-algebra $\mathcal{S}^*$ generated by a NIS $\mathcal{S}$ satisfies the axioms $(N_1) - (N_{16})$, and hence, every NIS-algebra generated by the NISs are abstract NIS-algebra.

Example 14.1. Let us consider the NIS $\mathcal{S}$ of Table 14.4, which is a modified form of the one given in [3].

Table 14.4. NIS $\mathcal{S}$

	Languages (L)	Sports (S)
Ann	{Arabic, Bulgarian}	{athletics, basketball}
Bob	{Arabic, Dutch}	{athletics, basketball}
Cindy	{German, Dutch}	{cycling}

Here $\mathcal{A} := \{L, S\}$, $\mathcal{V}_L := \{$Arabic, Bulgarian, Dutch, German$\}$, and $\mathcal{V}_S := \{$athletics, basketball, cycling$\}$. Thus, $|\mathcal{D}^L| = 16$ and $|\mathcal{D}^S| = 8$. Let us take $\Theta := \{1, 2, \ldots, 128\}$. A NIS-algebra generated by NIS $\mathcal{S}$ is given by

$$\mathcal{S}^* := (\wp(U), \cap, \sim, \emptyset, \{\underline{Ind_B^{\mathcal{S}}}\}_{B\subseteq\mathcal{A}}, \{\underline{Sim_B^{\mathcal{S}}}\}_{B\subseteq\mathcal{A}}, \{\underline{In_B^{\mathcal{S}}}\}_{B\subseteq\mathcal{A}}, \{c_\gamma^{\mathcal{S}}\}_{\gamma\in\mathcal{D}}, \{c_i^{\mathcal{S}}\}_{i\in\Theta}),$$

where $U :=$ {Ann, Bob, Cindy}, and the operators are given by Tables 14.5 and 14.6.
We note that

$$c^{\mathcal{S}}_{(\text{L,Bulgarian})} \not\subseteq \underline{Sim^{\mathcal{S}}_{\{L\}}}(c^{\mathcal{S}}_{(\text{L,Bulgarian})}), \tag{14.7}$$

$$c^{\mathcal{S}}_{(\text{L,Arabic})} \cap \underline{In^{\mathcal{S}}_{\{L,S\}}}(\{Bob\}) \not\subseteq \underline{In^{\mathcal{S}}_{\{S\}}}(c^{\mathcal{S}}_{(\text{L,Arabic})} \rightarrow \{Bob\}), \tag{14.8}$$

$$c^{\mathcal{S}}_{(\text{L,Arabic})} \cap \underline{Ind^{\mathcal{S}}_{\{L,S\}}}(\{Bob\}) \not\subseteq \underline{Ind^{\mathcal{S}}_{\{S\}}}(c^{\mathcal{S}}_{(\text{L,Arabic})} \rightarrow \{Bob\}). \tag{14.9}$$

So (14.7)–(14.9) of Example 14.1 show that in a NIS, (N_2) may not be satisfied by the lower approximation operators corresponding to similarity relations and (N_6) may not be satisfied by the lower approximation operators corresponding to indiscernibility and inclusion relations. It is thus that we have used (N_4) as the replacement of (N_2) for the similarity relation, and (N_5), (N_7) as the replacements of (N_6)

Table 14.5. Lower approximation operators generated by the NIS $\mathcal{S}$

	$\underline{Ind^{\mathcal{S}}_{\{L\}}}$, $\underline{Ind^{\mathcal{S}}_{\{L,S\}}}$, $\underline{In^{\mathcal{S}}_{\{L\}}}$, $\underline{In^{\mathcal{S}}_{\{L,S\}}}$	$\underline{Sim^{\mathcal{S}}_{\{L\}}}$	$\underline{Ind^{\mathcal{S}}_{\{S\}}}$, $\underline{Sim^{\mathcal{S}}_{\{S\}}}$, $\underline{In^{\mathcal{S}}_{\{S\}}}$, $\underline{Sim^{\mathcal{S}}_{\{L,S\}}}$
{Ann}	{Ann}	$\emptyset$	$\emptyset$
{Bob}	{Bob}	$\emptyset$	$\emptyset$
{Cindy}	{Cindy}	$\emptyset$	{Cindy}
{Ann, Bob}	{Ann, Bob}	{Ann}	{Ann, Bob}
{Ann, Cindy}	{Ann, Cindy}	$\emptyset$	{Cindy}
{Bob, Cindy}	{Bob, Cindy}	{Cindy}	{Cindy}
U	U	U	U
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Table 14.6. Nullary operators $c^{\mathcal{S}}_{\gamma}$ and $c^{\mathcal{S}}_{i}$

(a) Nullary operators $c^{\mathcal{S}}_{\gamma}$

γ	$c^{\mathcal{S}}_{\gamma}$
(L,Arabic)	{Ann, Bob}
(L,Bulgarian)	{Ann}
(L,Dutch)	{Bob, Cindy}
(L,German)	{Cindy}
(S,athletics)	{Anna, Bob}
(S,basketball)	{Anna, Bob}
(S,cycling)	{Cindy}

(b) Nullary operators $c^{\mathcal{S}}_{i}$

$i \in \Theta$	$c^{\mathcal{S}}_{i}$
1	{Ann}
2	{Bob}
3	{Cindy}
$i \in \Theta \setminus \{1,2,3\}$	$\emptyset$

for indiscernibility and inclusion relations, respectively. Observe that these replacements make use of the unary operators d_i.

14.7 Representation Theorem for Abstract NIS-Algebras and Equational Logic for NISs

In this section, we shall prove the representation theorem for abstract NIS-algebras, which will also lead us to an equational logic for NISs. We proceed as in the case of abstract DIS-algebras and for each $B \subseteq \mathcal{A}$, we consider the relations $L^{\mathfrak{A}}_{B} \subseteq PF(\mathfrak{A}) \times PF(\mathfrak{A})$, $L \in \{I,S,N\}$, generated by an abstract NIS-algebra

$$\mathfrak{A} := (U, \wedge, \neg, 0, \{I_B\}_{B\subseteq\mathcal{A}}, \{S_B\}_{B\subseteq\mathcal{A}}, \{N_B\}_{B\subseteq\mathcal{A}}, \{d_\gamma\}_{\gamma\in\mathcal{D}}, \{d_i\}_{i\in\Theta}),$$

such that

$$(\Gamma,\Delta) \in L^{\mathfrak{A}}_{B} \text{ if and only if } L_B(x) \in \gamma \text{ implies } x \in \Delta.$$

For a $\Gamma \in PF(\mathfrak{A})$, and $b \in \mathcal{A}$, let Γ_b denote the set $\{d_{(b,v)} : d_{(b,v)} \in \Gamma\}$. The following proposition presents a few properties of these operators.

Proposition 14.5

1. $L^{\mathfrak{A}}_B \subseteq L^{\mathfrak{A}}_C$ *for* $C \subseteq B \subseteq \mathcal{A}$, $L \in \{I,S,N\}$.
2. $\Gamma_b = \Delta_b$ *if and only if* $(\Gamma,\Delta) \in I^{\mathfrak{A}}_{\{b\}}$.
3. $\Gamma_b \cap \Delta_b \neq \emptyset$ *if and only if* $(\Gamma,\Delta) \in S^{\mathfrak{A}}_{\{b\}}$.
4. $\Gamma_b \subseteq \Delta_b$ *if and only if* $(\Gamma,\Delta) \in N^{\mathfrak{A}}_{\{b\}}$.
5. *If* $(\Gamma,\Delta) \in I^{\mathfrak{A}}_B$ *and* $\Gamma_b = \Delta_b$, *then* $(\Gamma,\Delta) \in I^{\mathfrak{A}}_{B\cup\{b\}}$.
6. *If* $(\Gamma,\Delta) \in S^{\mathfrak{A}}_B$ *and* $\Gamma_b \cap \Delta_b \neq \emptyset$, *then* $(\Gamma,\Delta) \in S^{\mathfrak{A}}_{B\cup\{b\}}$.
7. *If* $(\Gamma,\Delta) \in N^{\mathfrak{A}}_B$ *and* $\Gamma_b \subseteq \Delta_b$, *then* $(\Gamma,\Delta) \in N^{\mathfrak{A}}_{B\cup\{b\}}$.
8. $L^{\mathfrak{A}}_{\emptyset} = PF(\mathfrak{A}) \times PF(\mathfrak{A})$, $L \in \{I,S,N\}$.
9. $L^{\mathfrak{A}}_B = \bigcap_{b\in B} L^{\mathfrak{A}}_{\{b\}}$, $L \in \{I,S,N\}$.

Proof We only provide the proofs of (2) and (3).
(2) First suppose $\Gamma_b = \Delta_b$, and let $I_{\{b\}}(x) \in \Gamma$. We need to show $x \in \Delta$. From (N_{12}), we obtain $d_i \in \Gamma$ for some $i \in \Theta$. Therefore, $d_i \wedge I_{\{b\}}(x) \in \Gamma$. Now using (N_5) for $B = \emptyset$, we obtain

$$I_\emptyset\left(\bigwedge_{v\in\mathcal{V}_b}\left(d_{(b,v)} \leftrightarrow I_\emptyset(d_i \to d_{(b,v)})\right) \to x\right) \in \Gamma.$$

Therefore, (N_9) gives

$$\bigwedge_{v\in\mathcal{V}_b}\left(d_{(b,v)} \leftrightarrow I_\emptyset(d_i \to d_{(b,v)})\right) \to x = 1,$$

and hence

$$\bigwedge_{v\in\mathcal{V}_b}\left(d_{(b,v)} \leftrightarrow I_\emptyset(d_i \to d_{(b,v)})\right) \to x \in \Delta.$$

If possible, let $x \notin \Delta$. Then, there exists a $v \in \mathcal{V}_b$ such that either $d_{(b,v)} \to I_\emptyset(d_i \to d_{(b,v)}) \notin \Delta$, or $I_\emptyset(d_i \to d_{(b,v)}) \to d_{(b,v)} \notin \Delta$. First suppose, $d_{(b,v)} \to I_\emptyset(d_i \to d_{(b,v)}) \notin \Delta$. Then, $d_{(b,v)} \in \Delta$, and $I_\emptyset(d_i \to d_{(b,v)}) \notin \Delta$. Now using the fact that $\Gamma_b = \Delta_b$, we obtain $d_{(b,v)} \in \Gamma$, and hence, $d_i \wedge d_{(b,v)} \in \Gamma$. Therefore, from $(N_{10}$, we obtain $I_\emptyset(d_i \to d_{(b,v)}) \in \Gamma$. Using (N_9), this gives $d_i \to d_{(b,v)} = 1$, and hence, by (N_{15}), $I_\emptyset(d_i \to d_{(b,v)}) = 1$. This implies $I_\emptyset(d_i \to d_{(b,v)}) \in \Delta$, a contradiction. Similarly, $I_\emptyset(d_i \to d_{(b,v)}) \to d_{(b,v)} \notin \Delta$ will also lead us to a contradiction.

Conversely, suppose $(\Gamma,\Delta) \in I^{\mathfrak{A}}_{\{b\}}$. We need to show $d_{(b,v)} \in \Gamma$ if and only if $d_{(b,v)} \in \Delta$. First let $d_{(b,v)} \in \Gamma$. Then, from (N_2), we obtain $I_{\{b\}}d_{(b,v)} \in \Gamma$, and hence $d_{(b,v)} \in \Delta$. Now suppose $d_{(b,v)} \in \Delta$. If $d_{(b,v)} \notin \Gamma$, then using (N_3), we obtain $I_{\{b\}}(\neg d_{(b,v)}) \in \Gamma$, and hence, $\neg d_{(b,v)} \in \Delta$, a contradiction.

(3) First suppose $\Gamma_b \cap \Delta_b \neq \emptyset$ and we show $(\gamma,\Delta) \in S^{\mathfrak{A}}_{\{b\}}$. Let $d_{(b,v)} \in \gamma_b \cap \Delta_b$. Let $S_{\{b\}}(x) \in \gamma$. We need to show $x \in \Delta$. We have $d_{(b,v)} \wedge S_{\{b\}}(x) \in \Gamma$, and hence by

(N_6), $S_\emptyset(d_{(b,v)} \to x) \in \Gamma$. This gives $I_\emptyset(d_{(b,v)} \to x) \in \Gamma$ (due to (N_8)), and therefore, we obtain $d_{(b,v)} \to x = 1$ (by (N_9)). This gives us $x \in \Delta$ as $d_{(b,v)} \in \Delta$.
Conversely, suppose $(\Gamma, \Delta) \in S^{\mathfrak{A}}_{\{b\}}$, and we prove $\Gamma_b \cap \Delta_b \neq \emptyset$. (N_{12}) guarantees the existence of a $i \in \Theta$ such that $d_i \in \Gamma$. Therefore, by (N_4), we obtain $S_{\{b\}}(\bigvee_{v \in \mathcal{V}_b}(d_{(b,v)} \wedge I_\emptyset(d_i \to d_{(b,v)}))) \in \Gamma$, and hence, $\bigvee_{v \in \mathcal{V}_b}(d_{(b,v)} \wedge I_\emptyset(d_i \to d_{(b,v)})) \in \Delta$. Therefore, for some $v \in \mathcal{V}_b$, $d_{(b,v)} \wedge I_\emptyset(d_i \to d_{(b,v)}) \in \Delta$. Using (N_9) and the fact that $d_i \in \Gamma$, this gives us $d_{(b,v)} \in \gamma \cap \Delta$, and hence, $\Gamma_b \cap \Delta_b \neq \emptyset$. □

Let us consider an abstract NIS-algebra $\mathfrak{A} := (U, \wedge, \neg, 0, \{I_B\}_{B \subseteq \mathcal{A}}, \{S_B\}_{B \subseteq \mathcal{A}}, \{N_B\}_{B \subseteq \mathcal{A}}, \{d_\alpha\}_{\alpha \in \mathcal{D}}, \{d_i\}_{i \in \Theta})$. As in the case of abstract DIS algebra, abstract NIS-algebra $\mathfrak{A}$ determines a unique NIS $\mathfrak{A}_* := (PF(\mathfrak{A}), \mathcal{A}, \mathcal{V}, f_{\mathfrak{A}})$, where

$$f_{\mathfrak{A}}(\Gamma, a) = \{v : d_{(a,v)} \in \Gamma\}.$$

The following proposition relates the abstract NIS-algebra $\mathfrak{A}$ and NIS $\mathfrak{A}_*$.

Theorem 14.4. *The following hold for each $B \subseteq \mathcal{A}$.*

1. a. $Ind_B^{\mathfrak{A}_*} = I_B^{\mathfrak{A}}$.
   b. $Sim_B^{\mathfrak{A}_*} = S_B^{\mathfrak{A}}$.
   c. $In_B^{\mathfrak{A}_*} = N_B^{\mathfrak{A}}$.
2. a. $\underline{Ind_B^{\mathfrak{A}_*}} = m_{I_B^{\mathfrak{A}}}$.
   b. $\underline{Sim_B^{\mathfrak{A}_*}} = m_{S_B^{\mathfrak{A}}}$.
   c. $\underline{In_B^{\mathfrak{A}_*}} = m_{N_B^{\mathfrak{A}}}$.

Proof. We only prove (1) for singleton B. First suppose $(\Gamma, \Delta) \in Ind^{\mathfrak{A}_*}_{\{b\}}$. Therefore, we obtain $f_{\mathfrak{A}}(\Gamma, b) = f_{\mathfrak{A}}(\Delta, b)$. This implies $\Gamma_b = \Delta_b$, and hence, by item (2) of Proposition 14.5, we obtain $(\Gamma, \Delta) \in I^{\mathfrak{A}}_{\{b\}}$, as desired. Conversely, let $(\Gamma, \Delta) \in I^{\mathfrak{A}}_{\{b\}}$. Then by item (2) of Proposition 14.5, we obtain $\Gamma_b = \Delta_b$. This gives $f_{\mathfrak{A}}(\Gamma, b) = f_{\mathfrak{A}}(\Delta, b)$ and hence $(\Gamma, \Delta) \in Ind^{\mathfrak{A}_*}_{\{b\}}$. One can prove 1(b) and 1(c) in the same way. □

For each $i \in \Theta$, let us consider the nullary operators $c_i^{\mathfrak{A}_*}$ defined as

$$c_i^{\mathfrak{A}_*} := \{\Gamma \in PF(\mathfrak{A}) : d_i \in \Gamma\}.$$

Then we obtain the following theorem.

Theorem 14.5

1. $PF(\mathfrak{A})/Ind_{\mathcal{A}}^{\mathfrak{A}_*} \subseteq \{c_i^{\mathfrak{A}_*} : i \in \Theta\}$.
2. $c_i^{\mathfrak{A}_*} \cap c_j^{\mathfrak{A}_*} = \emptyset$, *for* $i \neq j$.
3. $c_i^{\mathfrak{A}_*} \in U/Ind_{\mathcal{A}}^{\mathfrak{A}_*} \cup \{\emptyset\}$.
4. *If* $(\Gamma, \Delta) \notin Ind_{\mathcal{A}}^{\mathfrak{A}_*}$, $[\Gamma]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} = c_i^{\mathfrak{A}_*}$ *and* $[\Delta]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} = c_j^{\mathfrak{A}_*}$, *then* $i \neq j$.

Proof (1) Let $[\Gamma]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} \in U/Ind_{\mathcal{A}}^{\mathfrak{A}_*}$. From (N_{12}), we obtain $d_i \in \Gamma$ for some $i \in \Theta$. Also, due to (N_{13}), $d_j \notin \Gamma$ for all j distinct from i. We claim that $[\Gamma]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} = c_i^{\mathfrak{A}_*}$. In order to see it, first suppose $(\Gamma,\Delta) \in Ind_{\mathcal{A}}^{\mathfrak{A}_*}$. We need to show $d_i \in \Delta$. But since $d_i \in \Gamma$, we obtain, using (N_{14}), $I_{\mathcal{A}} d_i \in \Gamma$. Now using the facts that $(\Gamma,\Delta) \in Ind_{\mathcal{A}}^{\mathfrak{A}_*}$ and $Ind_{\mathcal{A}}^{\mathfrak{A}_*} = I_{\mathcal{A}}^{\mathfrak{A}}$, we obtain $d_i \in \Delta$. Next, suppose $\Delta \in c_i^{\mathfrak{A}_*}$, and we prove $(\Gamma,\Delta) \in Ind_{\mathcal{A}}^{\mathfrak{A}_*}$. For this, by item (2) of Proposition 14.5, it is enough to show that for each $a \in \mathcal{A}$, and $v \in \mathcal{V}_a$, $d_{(a,v)} \in \Gamma$ if and only if $d_{(a,v)} \in \Delta$. First suppose $d_{(a,v)} \in \Gamma$. Then using (N_{10}), we obtain $I_{\emptyset}(d_i \to d_{(a,v)}) \in \Gamma$. Thus using (N_9), we obtain $d_i \to d_{(a,v)} \in \Delta$, and hence $d_{(a,v)} \in \Delta$, as $d_i \in \Delta$. Similarly, using (N_{11}), one can show that if $d_{(a,v)} \in \Delta$, then $d_{(a,v)} \in \Gamma$.

(2) Follows from (N_{13}).

(3) Let us consider $c_i^{\mathfrak{A}_*} \neq \emptyset$. Then there exists $\gamma \in c_i^{\mathfrak{A}_*}$. Now giving argument similar to (1), one can show that $[\Gamma]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} = c_i^{\mathfrak{A}_*}$.

(4) If possible, let $(\Gamma,\Delta) \notin Ind_{\mathcal{A}}^{\mathfrak{A}_*}$, $[\Gamma]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} = [\Delta]_{Ind_{\mathcal{A}}^{\mathfrak{A}_*}} = c_i^{\mathfrak{A}_*}$. Since $(\Gamma,\Delta) \notin Ind_{\mathcal{A}}^{\mathfrak{A}_*}$, without loss of generality we assume the existence of an $a \in \mathcal{A}$ and v such that $v \in f_{\mathfrak{A}}(\Gamma,a)$, but $v \notin f_{\mathfrak{A}}(\Delta,a)$. That is, $d_{(a,v)} \in \Gamma$ and $d_{(a,v)} \notin \Delta$. Therefore, we obtain $d_i \wedge d_{(a,v)} \in \Gamma$ and by (N_{10}), $I_{\emptyset}(d_i \to d_{(a,v)}) \in \Gamma$. This implies $I_{\emptyset}(d_i \to d_{(a,v)}) \neq 0$ and hence by (N_9), $d_i \to d_{(a,v)} = 1$. But this contradicts the fact that $d_i \wedge \neg d_{(a,v)} \in \Delta$. □

Finally, we consider the NIS algebra $(\mathfrak{A}_*)^*$ generated by $\mathfrak{A}_*$ by taking nullary operators corresponding to elements of Θ as $c_i^{\mathfrak{A}_*}$. That is,

$$(\mathfrak{A}_*)^* := (\wp(PF(\mathfrak{A})), \cap, \sim, \emptyset, \{\underline{Ind_B^{\mathfrak{A}_*}}\}_{B\subseteq\mathcal{A}}, \{\underline{Sim_B^{\mathfrak{A}_*}}\}_{B\subseteq\mathcal{A}}, \{\underline{In_B^{\mathfrak{A}_*}}\}_{B\subseteq\mathcal{A}}, \{c_\gamma^{\mathfrak{A}_*}\}_{\gamma\in\mathcal{D}}, \{c_i^{\mathfrak{A}_*}\}_{i\in\Theta}).$$

Now, one can prove the following representation theorem for abstract NIS-algebras, using Theorem 14.4.

Theorem 14.6 (Representation theorem for abstract NIS-algebras)
Let $\mathfrak{A} := (U, \wedge, \neg, 0, \{I_B\}_{B\subseteq\mathcal{A}}, \{S_B\}_{B\subseteq\mathcal{A}}, \{N_B\}_{B\subseteq\mathcal{A}}, \{d_\alpha\}_{\alpha\in\mathcal{D}}, \{d_i\}_{i\in\Theta})$ *be an abstract NIS-algebra. Then, the mapping* $\Psi : U \to \wp(PF(\mathfrak{A}))$ *given by*

$$\Psi(x) := \{\Gamma \in PF(\mathfrak{A}) : x \in \Gamma\},\ x \in U,$$

is an embedding of $\mathfrak{A}$ *into* $(\mathfrak{A}_*)^*$.

As in the case of DISs, the above representation theorem gives us an equational logic for NISs consisting of the axioms $(N_1) - (N_{16})$.

14.7.1 Extension to Other Types of Relations Defined on NISs

So far in our study of NISs, we have restricted ourselves to indiscernibility, similarity and inclusion relations. But, one can extend the scheme of this work to other

types of relations defined on NISs as well. In fact, as mentioned in Remark 14.2, the main task is to come up with axioms relating the approximations (with respect to the relation considered) relative to different sets of attributes and attribute-value pairs. We list below the axioms serving this purpose for the relations defined in Section 14.1. Let R_B, $B \subseteq \mathcal{A}$, be the operators corresponding to the relation considered.

Negative similarity relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_C(x) \leq R_B(x)$ for $C \subseteq B \subseteq \mathcal{A}$.
- $d_i \leq R_{\{a\}}(\bigvee_{v \in \mathcal{V}_a}(\neg d_{(a,v)} \wedge I_\emptyset(d_i \rightarrow \neg d_{(a,v)})))$.
- $\neg d_{(b,v)} \wedge R_{B \cup \{b\}}(x) \leq R_B(\neg d_{(b,v)} \rightarrow x)$.
- $R_\emptyset = I_\emptyset$.

Complementarity relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_C(x) \leq R_B(x)$ for $C \subseteq B \subseteq \mathcal{A}$.
- $d_{(a,v)} \leq R_{\{a\}}(\neg d_{(a,v)})$.
- $\neg d_{(a,v)} \leq R_{\{a\}}(d_{(a,v)})$.
- $d_i \wedge R_{B \cup \{b\}}(x) \leq R_B\left(\bigwedge_{v \in \mathcal{V}_b}\left(\neg d_{(b,v)} \leftrightarrow I_\emptyset(d_i \rightarrow d_{(b,v)})\right) \rightarrow x\right)$.
- $R_\emptyset = I_\emptyset$.

Weak indiscernibility relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_B(x) \leq R_C(x)$ for $C \subseteq B \subseteq \mathcal{A}$.
- $d_i \leq R_B\left(\bigvee_{b \in B}\bigwedge_{v \in \mathcal{V}_b}\left(d_{(b,v)} \leftrightarrow I_\emptyset(d_i \rightarrow d_{(b,v)})\right)\right)$.

Weak similarity relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_B(x) \leq R_C(x)$ for $C \subseteq B \subseteq \mathcal{A}$.
- $d_i \leq R_B\left(\bigvee_{b \in B}\bigvee_{v \in \mathcal{V}_b}\left(d_{(b,v)} \wedge I_\emptyset(d_i \rightarrow d_{(b,v)})\right)\right)$.

Weak inclusion relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_B(x) \leq R_C(x)$ for $C \subseteq B \subseteq \mathcal{A}$.
- $d_i \leq R_B\left(\bigvee_{b \in B}\bigwedge_{v \in \mathcal{V}_b}\left(I_\emptyset(d_i \rightarrow d_{(b,v)}) \rightarrow d_{(b,v)}\right)\right)$.

Weak negative similarity relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_B(x) \leq R_C(x)$ for $C \subseteq B \subseteq \mathcal{A}$.

- $d_i \leq R_B\left(\bigvee_{b\in B}\bigvee_{v\in V_b}\left(\neg d_{(b,v)} \wedge I_\emptyset(d_i \rightarrow \neg d_{(b,v)})\right)\right)$.

Weak complementarity relation:

- $R_B(x \wedge y) = R_B(x) \wedge R_B(y)$.
- $R_B(x) \leq R_C(x)$ for $C \subseteq B \subseteq \mathcal{A}$.
- $d_i \leq R_B\left(\bigvee_{b\in B}\bigwedge_{v\in V_b}\left(\neg d_{(b,v)} \leftrightarrow I_\emptyset(d_i \rightarrow d_{(b,v)})\right)\right)$.

Thus we need to consider the axioms listed above, in addition to the axioms (N_9), (N_{15}) for $I_\emptyset$, and (N_{10})–(N_{14}) for d_i. Then one can obtain the counterparts of Theorem 14.4 for these relations. As a consequence of this, we would obtain the desired representation theorems for abstract NIS-algebras that include any of the above relations.

14.8 Conclusions

Classes of algebras induced by information systems – deterministic, incomplete or non-deterministic, are considered. These algebras are also able to capture the notion of approximations defined on these information systems. Abstract algebras are proposed, which model such classes of algebras. Corresponding representation theorems are proved. The representation theorems also lead us to equational logics for the respective information systems. In the process, it is also established that the proposed classes of abstract algebras for DISs and ISs constitute the algebraic counterparts of the logics for information systems studied in [15, 16].

A search for a suitable logic for information systems and rough set approximations remains the main issue of many research articles (*e.g.,* [23, 28, 24, 25, 26, 30, 2, 1, 15, 16], c.f. [11, 5]). In [16], the logics for deterministic/incomplete information systems are extended to propose dynamic logics for information systems, which can capture a formalization of the notion of information and information update in the context of information systems. A natural question would be to extend this work and propose a dynamic logic for non-deterministic information systems. A first step in this direction would be to translate the equational logic for NISs obtained in this chapter, into a modal logic. An extension of the latter to a dynamic logic for NISs may then be thought of, where we can capture the notion of information flow and information update for NISs.

References

1. Balbiani, P.: Axiomatization of logics based on Kripke models with relative accessibility relations. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 553–578. Physica Verlag, Heidelberg (1998)

2. Balbiani, P., Orłowska, E.: A hierarchy of modal logics with relative accessibility relations. Journal of Applied Non-Classical Logics 9(2-3), 303–328 (1999)
3. Balbiani, P., Vakarelov, D.: A modal logic for indiscernibility and complementarity in information systems. Fundamenta Informaticae 50, 243–263 (1950)
4. Banerjee, M., Chakraborty, M.K.: Algebras from rough sets. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neuro Computing: Techniques for Computing with Words, pp. 157–184. Springer, Berlin (2004)
5. Banerjee, M., Khan, M. A.: Propositional Logics from Rough Set Theory. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI, Part I. LNCS, vol. 4374, pp. 1–25. Springer, Heidelberg (2007)
6. Birkhoff, G.: On the structure of abstract algebras. Proceedings of the Cambridge Philosophical Society 29, 441–464 (1935)
7. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
8. Comer, S.: An algebraic approach to the approximation of information. Fundamenta Informaticae XIV, 492–502 (1991)
9. Düntsch, I., Gediga, G., Orłowska, E.: Relational attribute systems. International Journal of Human Computer Studies 55(3), 293–309 (2001)
10. Düntsch, I., Gediga, G., Orłowska, E.: Relational Attribute Systems II: Reasoning with Relations in Information Structures. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII, Part II. LNCS, vol. 4400, pp. 16–35. Springer, Heidelberg (2007)
11. Demri, S., Orłowska, E.: Incomplete Information: Structure, Inference, Complexity. Springer, Heidelberg (2002)
12. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17, 191–200 (1990)
13. Henkin, L., Monk, J.D., Tarski, A.: Cylindric Algebras, Part I. North-Holland Pub. Co., Amsterdam (1971)
14. Iwiński, T.B.: Algebraic approach to rough sets. Bulletin of the Polish Academy of Sciences (Math) 35(9-10), 673–683 (1987)
15. Khan, M. A., Banerjee, M.: A Logic for Complete Information Systems. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS(LNAI), vol. 5590, pp. 829–840. Springer, Heidelberg (2009)
16. Khan, M.A., Banerjee, M.: Logics for information systems and their dynamic extensions. ACM Transactions on Computational Logic 12(4), 29 (2011)
17. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: a tutorial. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 3–98. Springer, Singapore (1999)
18. Konikowska, B.: A formal language for reasoning about indiscernibility. Bulletin of the Polish Academy of Sciences 35, 239–249 (1987)
19. Konikowska, B.: A logic for reasoning about relative similarity. Studia Logica 58, 185–226 (1997)
20. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112, 39–49 (1998)
21. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113, 271–292 (1999)
22. Lin, T.Y., Yao, Y.Y.: Neighborhoods system: measure, probability and belief functions. In: Tsumoto, S., Kobayashi, S., Yokomori, T., Tanaka, H., Nakamura, A. (eds.) Proceedings of the The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, November 6-8, pp. 202–207. The University of Tokyo, Tokyo (1996)

23. Orłowska, E.: Dynamic information system. Fundamenta Informaticae 5, 101–118 (1982)
24. Orłowska, E.: Logic of Indiscernibility Relations. In: Skowron, A. (ed.) SCT 1984. LNCS, vol. 208, pp. 177–186. Springer, Heidelberg (1985)
25. Orłowska, E.: Logic of nondeterministic information. Studia Logica 1, 91–100 (1985)
26. Orłowska, E.: Kripke semantics for knowledge representation logics. Studia Logica XLIX, 255–272 (1990)
27. Orłowska, E.: Rough set semantics for non-classical logics. In: Ziarko, W. (ed.) Rough Sets,Fuzzy Sets and Knowledge Discovery, Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD 1993), Workshops in Computing, Banff, Alberta, Canada, October 12-15, pp. 143–148. Springer, Heidelberg (1993)
28. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. Theoretical Computer Science 29, 27–39 (1984)
29. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11(5), 341–356 (1982)
30. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
31. Polkowski, L.: Rough Sets: Mathematical Foundations. Physica-Verlag, Heidelberg (2002)
32. Pomykała, J.A.: Approximation, similarity and rough constructions. ILLC prepublication series for computation and complexity theory CT-93-07, University of Amsderdam (1993)
33. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
34. Ślęzak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
35. Vakarelov, D.: Abstract characterization of some knowledge representation systems and the logic *NIL* of nondeterministic information. In: Jorrand, P., Sgurev, V. (eds.) Artificial Intelligence II, pp. 255–260. North–Holland (1987)
36. Vakarelov, D.: Modal logics for knowledge representation systems. Theoretical Computer Science 90, 433–456 (1991)
37. Wasilewski, L., Ślęzak, D.: Foundations of rough sets from vagueness perspective. In: Hassanien, A.E., Suraj, Z., Ślęzak, D., Lingras, P. (eds.) Rough Computing, Theories, Technologies and Applications, pp. 1–37. IGI Global, Hershey (2008)
38. Yao, Y.Y.: Generalized rough set models. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 286–318. Physica-Verlag, Heidelberg (1998)
39. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)

Chapter 15
DNA Rough-Set Computing in the Development of Decision Rule Reducts*

Ikno Kim, Junzo Watada, and Witold Pedrycz

Abstract. Rough set methods are often employed for reducting decision rules. The specific techniques involving rough sets can be carried out in a computational manner. However, they are quite demanding when it comes computing overhead. In particular, it becomes problematic to compute all minimal length decision rules while dealing with a large number of decision rules. This results in an NP-hard problem. To address this computational challenge, in this study, we propose a method of DNA rough-set computing composed of computational DNA molecular techniques used for decision rule reducts. This method can be effectively employed to alleviate the computational complexity of the problem.

Keywords: DNA computation, decision rule reduction, NP hard problem, digraph, DNA rough-set computing, encoding process, deoxyribonucleic acid, nitrogen-containing base, hydrogen bond, DNA molecular technique, restriction enzyme technique, ligation technique, polymerase chain reaction technique, affinity separation technique, gel electrophoresis technique

15.1 Introduction

Many different types of databases and information technologies are needed to arrive at new solutions and to provide meaningful results to new problems, especially those encountered in machine learning, data mining, and other information

Ikno Kim · Junzo Watada
Graduate School of Information, Production and Systems, Waseda University,
Kitakyushu 808-0135, Japan
e-mail: octoberkim@akane.waseda.jp, watada@waseda.jp

Witold Pedrycz
Department of Electrical and Computer Engineering, University of Alberta,
Edmonton T6R 2G7, Canada
e-mail: wpedrycz@ualberta.ca

* This book chapter has been prepared on a basis of the results of the paper, entitled "A DNA-based algorithm for minimizing decision rules: a rough sets approach" to be published in IEEE Transactions on NanoBioscience at the end of 2011.

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 409–438.
springerlink.com

processing tasks [30]-[21]. The derived results could include, for example, decisions, features, and relationships that are of interest to the user. With this regard, rough set methodology and algorithmic framework offers an interesting and useful alternative. In rough sets, decision tables composed of objects, attributes with their values, and decision attributes play a pivotal role. Rough sets provide a vehicle to realize numerous classification tasks. *Rough set-based methods* [18] are often applied to problems of feature solutions, meaning that certain specific characteristics of given selected objects and their attributes in each of the different decision classes are assessed and selected to provide simplified rules. Rough set theory forms an important basis of reasoning that generates understandable and transparent information that originates from large data sets. In rough sets, a decision table is often used to represent objects, attributes with values, and related decision attributes. Such a decision table should be computed to offer decision rule reducts, which can subsequently be interpreted in terms of a certain *if-then decision rule* set that is employed to support decision-making processes.

There have been great deals of research studies on rough sets and their applications. Those include some generalizations as proposed *for example*, by Słowiński [31]. A group of applications were focused on data analysis and building minimal-length decision rules and dealing with uncertainties. Grzymała-Busse [8] developed a computational method of decision rules of minimal length to manage uncertainty of data in expert systems. Subsequently, Ziarko [36] proposed a new type of decision system matrix to offer a novel way of reducting decision rules. In addition, Skowron and Rauszer [27] proposed an incremental identification algorithm based on a decision system matrix applied to a discernibility matrix. Another incremental algorithm was studied by Shan and Ziarko [28], who formed decision rules in a different way.

A large number of different types of decision rules come with several meaningful features (corresponding to objects, attributes with their values, and decision attributes). The most difficult problem that still remains in rough set theory is about how to reduct decision rules. Computing all minimal-length decision rules is an intractable NP-hard problem [24].

In this study, we propose a rough set method that is composed of computational DNA molecular techniques, including experimental molecular techniques, to reduct decision rules. This novel method is referred to as *DNA rough-set computing*. A DNA rough-set computing method is employed first to determine all of the low approximation subsets, which are classified into each of the selected decision classes, and secondly one derives decision rules of minimal length. Obviously, DNA rough-set computing was created based on the mathematical concepts of traditional rough sets and was carried out in this study to assess its efficiency in reducting decision rules. This demonstration of the DNA rough-set computing method provides the framework for building a new type of the rough set-based method. The material is organized as follows:

Section 15.2 describes some fundamentals of DNA molecules that are crucial to better understanding of the proposed method. Section 15.3 covers several relevant DNA molecular techniques. Section 15.4 brings some ideas about basic concept of rough sets as well as about a model decision table. Section 15.5 describes a novel

DNA rough-set computing method proposed in this study. Experimental studies are presented in Section 15.6. Finally, Section 15.7 includes some concluding comments and offers suggestions for further research.

15.2 Deoxyribonucleic Acid

Biological polymer material present in the cells of living organisms corresponds to *deoxyribonucleic acid*, which is commonly abbreviated as DNA [16]. DNA molecules perform various meaningful functions relevant to development and maintenance of living organisms. The main advantage of investigating the functions of DNA molecules from the computing perspective is the ability of the molecules to act as high volume memories capable of storing data for long periods of time while realizing such memories at the nanometer scale. In this section, we briefly describe DNA molecules, also referred to as *nitrogen-containing bases*, and explain two important natural bonds present in DNA molecules.

15.2.1 Nitrogen-Containing Bases

At the basic level, one single DNA molecule consists of phosphoric acid, a five-carbon sugar, and four different types of nitrogen-containing bases: *adenine* (A), *thymine* (T), *guanine* (G), and *cytosine* (C). Each of the four nitrogen-containing bases is shown in Fig. 15.1 along with their structural formula. Figs. 15.1(a) and (c),

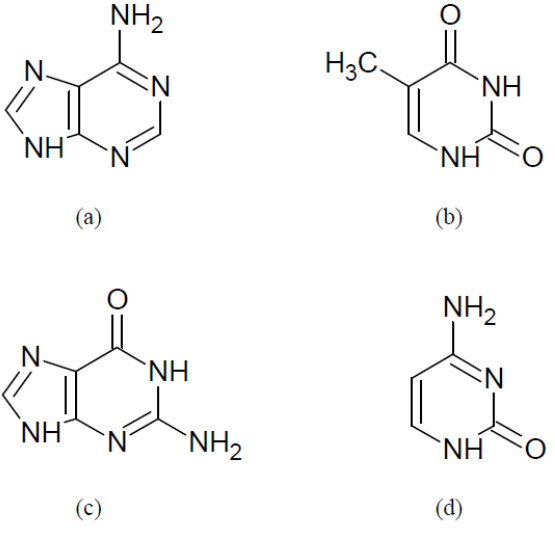


Fig. 15.1. Four nitrogen-containing bases in DNA represented as their structural formulas: (a) adenine (A) (double-ring structure); (b) thymine (T) (single-ring structure); (c) guanine (G) (double-ring structure); (d) cytosine (C) (single-ring structure)

respectively show the bases A and G, both of which have a double-ring structure and are called *purines*. Figs. 15.1(b) and (d), respectively show the bases T and C, both of which have a single-ring structure and are called *pyrimidines* [9]. In computational DNA molecular techniques, we must effectively manipulate these four nitrogen-containing bases A, T, G, and C to achieve meaningful processing.

15.2.2 Phosphodiester Bonds

Each of the four nitrogen-containing bases can be linked with one another by taking the sugar deoxyribose. If deoxyribose is attached to form a phosphate group, then it becomes a nucleotide. In order for nucleotides to join a polynucleotide chain, the phosphate is attached to the first sugar of the 5 carbon that is linked with the hydroxyl group attached to the second sugar of the 3 carbon. Through these types of chemical attachments, adjacent nucleotides of the sugars linked in component connections through the phosphates forming *phosphodiester bonds* [9]. Additionally, chemical compounds with phosphodiester bonds are built by nonmetal elements.

15.2.3 Hydrogen Bonds

In DNA molecules, two different polynucleotide strands are bound together within a pure liquid and are formed using shared hydrogen atoms of two negative atoms. The connection between the two single strands of polynucleotide is built by a bonding group referred to as a *hydrogen bond*. In particular, hydrogen bonds play a primary role in building the higher-order structure of the DNA molecule, such as an important formation of a right-handed helix. Hydrogen bonds form with observation of certain bonding patterns, which occur through normal types of base pairs (bp). That is, the base A is always attached only to the base T, whereas G is always attached only to C. This particular phenomenon is referred to as *Watson-Crick complementarity* [34]. Fig. 15.2 shows a photograph of the DNA molecule reconstruction model first built by Watson and Crick in 1953.

Because a purine chain is always attached only to a pyrimidine chain, the base dimensions become the same as the length of the DNA molecule. Thus, any of the two purines can selectively bond with any of the two pyrimidines. In other words, one complementary base pairing is composed of both A and T, and the other complementary base pairing is composed of both G and C. From these two complementary base pairings, all the backbone groups of the sugar-phosphate clearly have set orientations, which ultimately allow DNA molecules to presume the same structure for any bases of the DNA sequences. Importantly, the formation of base pairs is a symmetric process, which allows formation of the right-handed helix in double-stranded DNA (dsDNA). Hence, the symmetry of DNA molecules is related to the two single

polynucleotide strands, each of which runs exactly in opposite directions to create a dsDNA helix [35], [11].

Two different types of complementary bases should hold A and T together in two hydrogen bonds, as shown with their structural formulas in Fig. 15.3(a). Furthermore, two different types of complementary bases should hold G and C together in three hydrogen bonds, as shown with their structural formulas in Fig. 15.3(b). The combination of strong attachment in phosphodiester bonds and weak attachment in hydrogen bonds essentially makes the characteristic DNA molecule structure form.

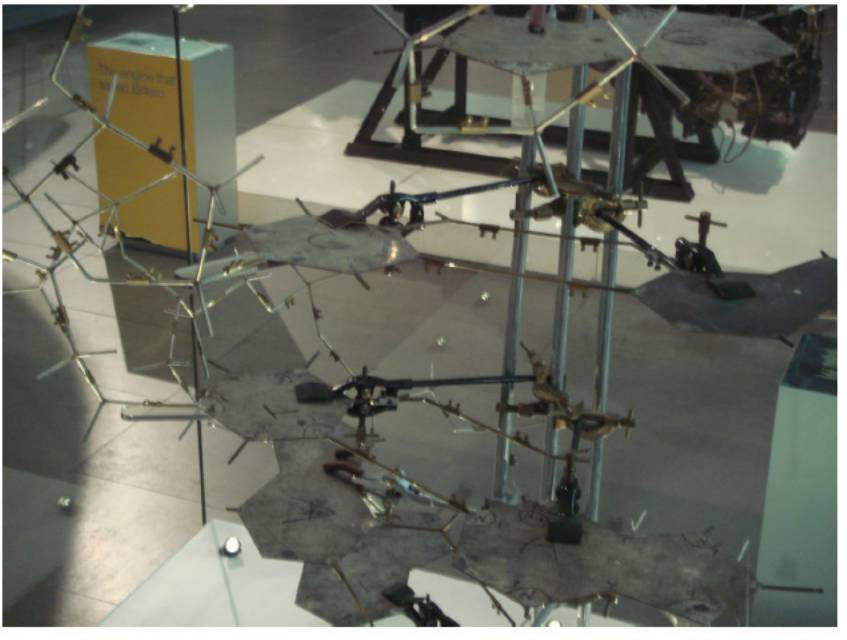


Fig. 15.2. DNA molecule reconstruction model built by Watson and Crick in 1953. The photograph was taken by Ikno Kim at Science Museum London (Copyright © 2011 by Ikno Kim)

15.3 DNA Molecular Techniques

In this section, we describe several important DNA molecular techniques to support a better understanding of how to manipulate DNA molecules in order to develop DNA rough-set computing. In particular, this section details five different types of DNA molecular techniques, which are the restriction enzyme technique, the ligation

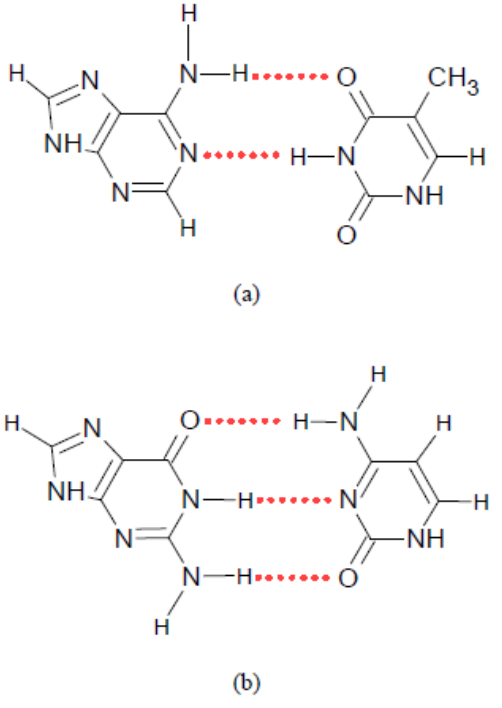


Fig. 15.3. Hydrogen bonds in Watson-Crick complementarity represented as their structural formulas: (a) adenine (A) and thymine (T) are two different complementary bases in two hydrogen bonds; (b) guanine (G) and cytosine (C) are two different complementary bases in three hydrogen bonds

technique, the polymerase chain reaction (PCR) technique, the affinity separation technique, and the gel electrophoresis technique. Each of these molecular techniques has its own significant function for manipulating DNA molecules and as such it becomes necessary for building robust DNA rough-set computing.

15.3.1 Restriction Enzyme Technique

To generate recombinant DNA, it is necessary both to break the desired DNA molecule sites and to isolate certain specific DNA fragments, which can be achieved by a DNA molecular technique referred to as the *restriction enzyme technique*. This technique intentionally severs DNA molecules into separate pieces by targeting certain sequences of the DNA. Such restriction enzyme sites are commonly comprised of four to six nucleotides, which involve their complementary DNA base parts [10]. DNA fragments of different lengths are generated by the amplification of various

types of DNA base sequences and by the subsequent breakage of them into separate pieces using the restriction enzyme technique. In other words, a restriction enzyme recognizes some specific DNA base sequences and breaks them at a desired position.

Restriction enzymes act as nucleases, which also break at a desired position of a DNA internal edge (type II restriction endonucleases). Since the four possible bases (A, T, G, and C) are considered, the expected frequency achieved in any DNA base sequences can be calculated by 4^n, where n is the recognized sequence length, and the lengths of three alternative sites can be predicted. First, each of the 256-base pairs indicates tetranucleotide sites. Second, each of the 1024-base pairs indicates pentanucleotide sites. Finally, each of the 4096-base pair indicates hexanucleotide sites. Furthermore, the cleaving mechanisms carried out in the restriction enzyme technique are either sticky-ended or blunt-ended [32]. Fig. 15.4 illustrates an example of the mechanism of a restriction enzyme (*Eco*RI).

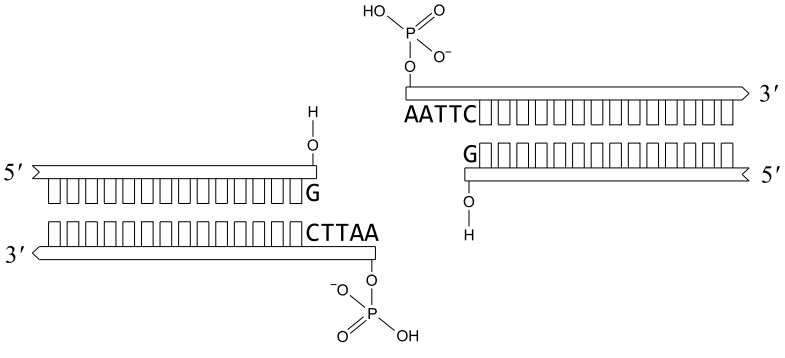


Fig. 15.4. Example mechanism of the restriction enzyme (*Eco*RI) with DNA base sequences of breaking sites, representing 5'-G|AATTC-3' and 5'-G|AATTC-3'

For the restriction enzyme technique, we add one or more enzymes with a buffer to DNA molecules. Certain restriction enzymes only work with a particular buffer that has a dissimilar salt concentration to ensure production of an efficient and trustworthy result. Most common restriction enzymes can react and be incubated at 37 °C. Basically, the amount of a restriction endonuclease used for a reaction (1 μg) should be able to completely digest substrate DNA in a volume of about 0.05 ml, keeping in mind three optimal settings, which are potential hydrogen (pH), temperature for one hour, and salt concentration [17].

15.3.2 Ligation Technique

Unstable dsDNA or dsDNA where one single DNA strand or two different opposite single DNA strands contain one or more DNA nucleotides and are not properly bound together can be repaired using a *ligation technique*. The attachment formation of phosphodiester bonds should be catalyzed by ligation for many important purposes in the recombinant DNA process.

DNA plasmid vectors are fused to create a single DNA molecule by a biochemical procedure, which is the *ligation process*. T4 ligations [7] can be used to seal the gaps in the single DNA strand for catalyzing the phosphodiester bonds. In the ligation process, T4 ligation works for both sticky-ended and blunt-ended types. The ligation buffer with enzyme that is used for the process involves adenosine triphosphate (ATP) that is labile at common laboratory room temperature [2]. All of the enzyme varieties and buffers used should be maintained on ice after they have been taken from the freezer.

15.3.3 Polymerase Chain Reaction Technique

PCR, a DNA molecular technique performed *in vitro*, can be used to amplify specific DNA sequences to a count of approximately a half a million by employing the use of simultaneous complementary primers. The PCR technique uses two types of primer extensions, which limit the region of DNA molecules for their amplifications. Subsequently, the PCR technique repeats the template-specific synthesis reaction of the DNA base sequences by inclusion of a heat-resistant polymerase. The PCR technique was developed to analyze the characteristics of DNA.

The PCR technique requires energy and nucleosides to synthesize DNA molecules with buffers, primers, polymerases, and templates. Particularly, polymerases, which are naturally occurring enzymes, are central components because they are used to repair and catalyze formation of DNA molecules [26]. The main step of synthesizing DNA molecules should be repeated and involves heating the latest synthesized DNA molecules until they are clearly separated and subsequently letting them cool to allow for primer formation, which anneals the DNA complementary sequences. With each cycle of heating and cooling, the DNA molecule number largely increases due to the primer extensions. The predominant reactions generate DNA molecule products, and some products are flanked by the primer extensions after several cycles. Fig. 15.5 shows a photograph of the commonly used type of PCR.

The DNA molecule count continues increasing during the heating and cooling cycles, although the enzyme might not synthesize an adequate desired number of DNA molecules or the reaction might suddenly decrease in intensity. The number of cycles required for optimal amplification is quite variable and is dependent on each of the amplification steps or on the initial amount of materials included.

Fig. 15.5. Commonly used type of PCR shows the substeps for each cycle. The photograph was taken by Ikno Kim at Taipei Medical University (Copyright © 2011 by Ikno Kim) (Courtesy of Prof. Jui-Yu Wu)

The PCR technique features should be established to maintain stability when the strong reactions are repeated at their high temperatures. Thus, the enzyme *Thermus aquaticus* (*Taq*) polymerase [15] should be used to maintain stability.

15.3.4 Affinity Separation Technique

To separate some desired DNA sequences from others or to extract one ore more specific DNA sequences, one DNA molecular technique that can be used is *affinity separation*. This technique plays a role of scales with a complicated mixture of DNA molecules from biochemical purpose extractions, cell homogenates, and fermentation broths [14].

The affinity separation technique often uses magnetic beads to collect and disperse material with an aqueous solution. This extraction technique is also employed to extract and separate biochemical matter. Another type of affinity separation technique uses magnetic beads with fluorescent labels to extract DNA molecules [6].

In this particular type of the affinity separation, fluorescently labeled DNA fragments have some specific base sequences of ligations that are attached to the magnetic beads. This type of affinity separation includes two functions, which are to enable the attachment to the magnetic beads and to identify each of the magnetic beads with a solidified DNA fragment. This type of separation technique is also able to deal with multiple DNA molecules extracted and tested at the same time. Many affinity separation techniques are expected to become automated ones.

15.3.5 Gel Electrophoresis Technique

The different length sizes of DNA molecules can be revealed by their different speeds as they move through an electric-charged gel in a technique called *gel electrophoresis*. DNA molecules are transferred into the electric-charged gel, and an electric current is introduced to the gel to make the DNA molecules separate according to their physical characteristics; the lengths of DNA molecules can be estimated using this technique. The common equipment used for the gel electrophoresis technique is called *agarose gel electrophoresis*. Gel electrophoresis is quite useful not only for measuring the DNA molecule length, but also for detaching the DNA fragments from their structures [16]. Fig. 15.6 shows a photograph of a commonly used type of agarose gel electrophoresis. A more recently introduced gel electrophoresis technique can measure DNA molecule length, while also analyzing the nucleic acids to provide molecular data. This newer apparatus automatically measures each single DNA strand and fragment and shows clear linear bands representing their lengths.

15.4 Rough Sets and a Model Decision Table

In this section, a fundamental concept of rough set theory is briefly revisited. This material is helpful in describing the DNA rough-set computing for decision rule reducts. A model decision table is also studied and is used as an example that will be solved by DNA rough-set computing.

15.4.1 Concept of Rough Set Theory

Rough set theory was introduced by Pawlak [19] based on a concept of the underlying set theory. He was the first to introduce numerous novel and innovative algorithms, such as fundamentals of Granular Computing, a novel formal computational machine, perceptional knowledge discovery, and others [23].

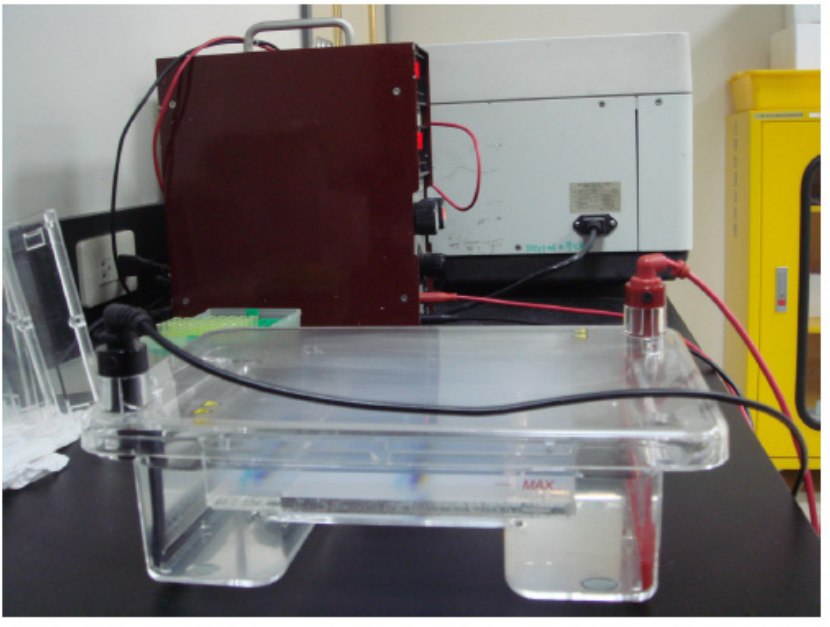


Fig. 15.6. Commonly used type of agarose gel electrophoresis shows the positive pole (red colour) and the negative pole (black colour). The photograph was taken by Ikno Kim at Taipei Medical University (Copyright © 2011 by Ikno Kim) (Courtesy of Prof. Jui-Yu Wu).

The fundamentals of rough set theory can be outlined as follows, see [25]. Assume that a universe is denoted by U, and a constraint set is denoted by T. T on objects is called a *disjoint relation* of object classifications. Assume that object x and object y (each object is an element in U), since (x,x) and (y,y) are elements of T and (x,y) need not be equal to (y,x). If three different elements x, y, and z are handled, T satisfies the properties of reflexivity, symmetry, and transitivity and is referred to as an *equivalence relation*. The equivalence relation, denoted by Ξ, is also called a *discernibility relation*. When this relation is invoked, all of the objects of U can be classified into three different types of disjoint sets: *lower approximation* (denoted by Ξ_A), the *upper approximation* (denoted by Ξ^A), and the *boundary region* (denoted by Ξ^B). Let us assume that X is a subset of U. The three disjoint sets for any subset $X \subseteq U$ express the multiplicity of objects, which are surely positioned in X (class 1), those that are not surely included in X (class 2), or those that are possibly located in X (class 3). The objects included in class 1 are from Ξ_A of X. The objects included in classes 1 and 3 are from Ξ^A of X. Finally, the objects included in class 3 are from Ξ^B of X [20].

Our objective is to determine the lower approximation subsets using DNA rough-set computing. The resultant lower approximation subsets are revealed by solving

the problem of the model decision table.The lower approximation subsets are determined before determining decision rule reducts using the DNA rough-set computing method.

15.4.2 Decision Table as a Model

In rough sets, we commonly deal with a decision table that is essentially created on a basis of a *decision system*. We define the decision system (denoted as *DS*) as $DS = (U, \Gamma, \varepsilon, \omega)$, where a finite condition attribute set is Γ, a decision attribute is ε, and a value assignment is ω. U consists of n objects $x_1, x_1, \cdots, x_n$, corresponding to $U = \{x_1, x_2, \cdots, x_n\}$. The n condition attributes are denoted as $\zeta_1, \zeta_2, \cdots, \zeta_n$, while Γ consists of n condition attributes, corresponding to $\Gamma = \{\zeta_1, \zeta_2, \cdots, \zeta_n\}$. The m condition attribute values are denoted as $\psi_1^{\zeta_1}, \psi_2^{\zeta_1}, \cdots, \psi_m^{\zeta_1}$, $\psi_1^{\zeta_2}, \psi_2^{\zeta_2}, \cdots, \psi_m^{\zeta_2}, \ldots$, and $\psi_1^{\zeta_n}, \psi_2^{\zeta_n}, \cdots, \psi_m^{\zeta_n}$ when we deal with condition attribute values for n condition attributes $\zeta_1, \zeta_2, \cdots, \zeta_n$. Finally, n pair sets of both n condition attributes and m condition attribute values are denoted as $I_{\zeta_1}^c, I_{\zeta_2}^c, \cdots, I_{\zeta_n}^c$.

A case has the mapping from $\varepsilon : U \to D^{\nu}$, where a set of decision values is D^{ν} in the decision system, where ω encompasses ε, representing $\omega : U \times (\Gamma \cup \{\varepsilon\}) \to I^{\nu} \cup D^{\nu}$, where a set of decision attributes is $\{\varepsilon\}$ also denoted as E, expressing $E = \{\varepsilon\}$, and this should be treated as a single decision attribute. Thus, a pair (ε, u) should belong to D^{ν}, meaning that it satisfies all of the value assignments. A set of all the attributes is expressed by $\Gamma_t = \Gamma \cup \{\varepsilon\}$ in the specific case, where the decision part should be added. Let us assume n decision attribute values are denoted as $\tau_1, \tau_2, \cdots, \tau_n$ and a decision attribute value set is expressed as $D^{\nu} = \{\tau_1, \tau_2, \cdots, \tau_n\}$. The n decision classes in the n-decision values are denoted as $D_{\tau_1}^{\nu}, D_{\tau_2}^{\nu}, \cdots, D_{\tau_n}^{\nu}$.

A model decision table was employed in this study for reducting decision rules in rough sets. The model decision table is composed of eight proposed objects, four proposed condition attributes with their condition attribute values, and a single decision attribute with their decision values. These given decision rules should be reducted to exhibit a minimal length. By reducting these decision rules, the characteristics of this decision table can be revealed. The given element sets for the model decision table are expressed as (1) U = {Object-1,Object-2,$\cdots$,Object-8}; (2) $\Gamma = \{*, \#, \blacksquare, \bullet\}$; (3) $I_*^c = \{(*,1),(*,2),(*,3),(*,4)\}$; (4) $I_\#^c = \{(\#,1),(\#,2),(\#,3),(\#,4)\}$; (5) $I_\blacksquare^c = \{(\blacksquare,1),(\blacksquare,2),(\blacksquare,3),(\blacksquare,4)\}$; (6) $I_\bullet^c = \{(\bullet,1),(\bullet,2),(\bullet,3)\}$; and (7) $D^{\nu} = \{$Decision Class-1, Decision Class-2$\}$. In more detail, the model decision table is composed of (1) Object-1 = $\{(*,4),(\#,3),(\blacksquare,2),(\bullet,1)\}$; (2) Object-2 = $\{(*,3),(\#,4),(\blacksquare,4),(\bullet,2)\}$; (3) Object-3 = $\{(*,1),(\#,1),(\blacksquare,1),(\bullet,3)\}$; (4) Object-4 = $\{(*,4),(\#,3),(\blacksquare,3),(\bullet,2)\}$; (5) Object-5 = $\{(*,2),(\#,2),(\blacksquare,1),(\bullet,1)\}$; (6) Object-6 = $\{(*,3),(\#,4),(\blacksquare,4),(\bullet,2)\}$; (7) Object-7 = $\{(*,4),(\#,1),(\blacksquare,2),(\bullet,3)\}$; (8) Object-8 = $\{(*,2),(\#,2),(\blacksquare,1),(\bullet,1)\}$; (9) Decision Class-1 = {Object-3, Object-4, Object-6, Object-8}; and (10) Decision Class-2 = {Object-1, Object-2, Object-5, Object-7}.

15.5 DNA Rough-Set Computing

DNA rough-set computing is again used to first determine the lower approximation subsets classified into each decision class and to determine decision rule reducts based on the determined results obtained so far. This section describes DNA rough-set computing as well as we show how the method is structuralized in order to reduct decision rules.

15.5.1 Digraph in DNA

The three distinctively different elements are object elements, condition attribute elements, and condition attribute value elements. To create a digraph of the decision table, we form both a condition attribute element and a condition attribute value element that become one integrated single element, referred to as a *pair element*. Thus, one pair element involves a condition attribute element and a condition attribute value element. By applying the two different elements, corresponding to object elements and pair elements, we create a new type of digraph composed of these two different elements, referred to as a *DNA-digraph*, as shown in Fig. 15.7. Hereafter, an object element and a pair element are called an *object node* and a *pair node* in the case of dealing with the DNA-digraph.

For the DNA-digraph, as for the decision table, let us assume that n object nodes are $x_1, x_2, \cdots, x_n$ and an object node set is U, expressed as $U = \{x_1, x_2, \cdots, x_n\}$, and n pair nodes consist of both n condition attributes $\zeta_1, \zeta_2, \cdots, \zeta_n$ and m condition attribute values $\psi_1^{\zeta_1}, \psi_2^{\zeta_1}, \cdots, \psi_m^{\zeta_1}, \psi_1^{\zeta_2}, \psi_2^{\zeta_2}, \cdots, \psi_m^{\zeta_2}, \ldots$, and $\psi_1^{\zeta_n}, \psi_2^{\zeta_n}, \cdots, \psi_m^{\zeta_n}$, and a pair node set is denoted as P_s, expressed as a *pair matrix*. The structure of the DNA-digraph is composed of both n object nodes and n pair nodes. Here, there are one or more pair nodes, in which some condition attributes, including fewer condition attribute values than other given condition attributes; the specific symbol 'ϕ' is set to be used to describe any empty entry.

A DNA-digraph transformed from the model decision table is shown in Fig. 15.8, in which each of the five complicated pair nodes $(*,4), (\#,1), (\#,3), (\blacksquare,1)$, and $(\blacksquare,2)$ is associated with multiple directions. The model DNA-digraph is composed of the eight object nodes, representing $U = \{x_1, x_2, \cdots, x_8\}$, and the four condition attributes, representing $\zeta_1, \zeta_2, \cdots, \zeta_4$. Here, the three condition attributes (ζ_1, ζ_2, and ζ_3) involve the four condition attribute values, and the a single condition attribute (ζ_4) involves the three condition attribute values. A model pair matrix is mainly used when we encode DNA sequences of the given pair nodes in single-stranded DNA (ssDNA).

The existing relation between any two nodes in the DNA-digraph, as shown in Fig. 15.7, is denoted as e, which represents the existing relation of a directed line connected with those two nodes. The three different types of directed relations between two nodes are the following cases. The first case is when an object node x_i

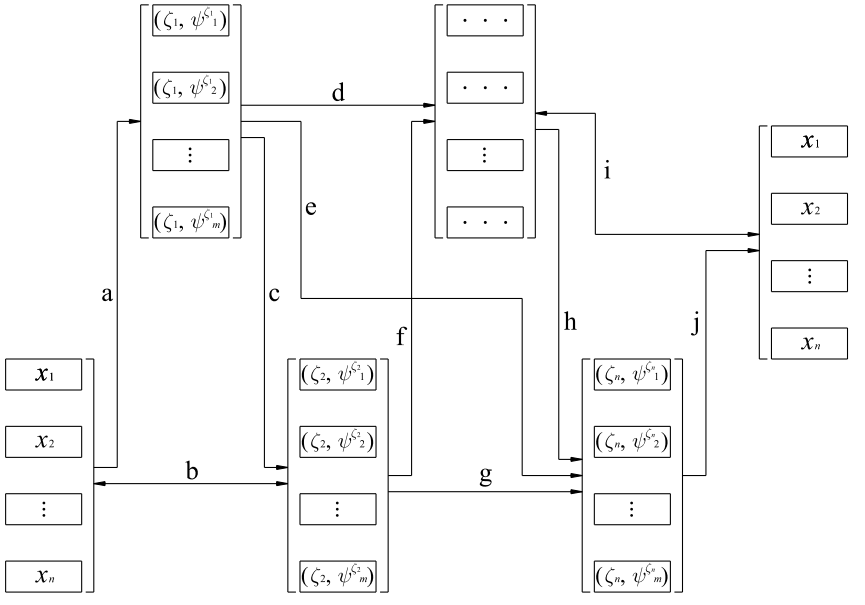


Fig. 15.7. Representation of a DNA-digraph is used to reduct decision rules. In this DNA-digraph, (1) **a** represents the directions from each of n object nodes $x_1, x_2, \cdots, x_n$ to each of n pair nodes $(\zeta_1, \psi_1^{\zeta_1})$, $(\zeta_1, \psi_2^{\zeta_1})$,$\cdots$, $(\zeta_1, \psi_m^{\zeta_1})$; (2) **b** represents the two-opposite-way directions from and to each of n object nodes $x_1, x_2, \cdots, x_n$ to and from each of n pair nodes $(\zeta_2, \psi_1^{\zeta_2})$, $(\zeta_2, \psi_2^{\zeta_2})$,$\cdots$, $(\zeta_2, \psi_m^{\zeta_2})$; (3) **c** represents the directions from each of n pair nodes $(\zeta_1, \psi_1^{\zeta_1})$, $(\zeta_1, \psi_2^{\zeta_1})$,$\cdots$, $(\zeta_1, \psi_m^{\zeta_1})$ to each of n pair nodes $(\zeta_2, \psi_1^{\zeta_2})$, $(\zeta_2, \psi_2^{\zeta_2})$,$\cdots$, $(\zeta_2, \psi_m^{\zeta_2})$; (4) **d** represents the directions from each of n pair nodes $(\zeta_1, \psi_1^{\zeta_1})$, $(\zeta_1, \psi_2^{\zeta_1})$,$\cdots$, $(\zeta_1, \psi_m^{\zeta_1})$ to each of n pair nodes $\cdots$, $\cdots$,$\ldots$, $\cdots$; (5) **e** represents the directions from each of n pair nodes $(\zeta_1, \psi_1^{\zeta_1})$, $(\zeta_1, \psi_2^{\zeta_1})$,$\cdots$, $(\zeta_1, \psi_m^{\zeta_1})$ to each of n pair nodes $(\zeta_n, \psi_1^{\zeta_n})$, $(\zeta_n, \psi_2^{\zeta_n})$,$\cdots$, $(\zeta_n, \psi_m^{\zeta_n})$; (6) **f** represents the directions from each of n pair nodes $(\zeta_2, \psi_1^{\zeta_2})$, $(\zeta_2, \psi_2^{\zeta_2})$,$\cdots$, $(\zeta_2, \psi_m^{\zeta_2})$ to each of n pair nodes $\cdots$, $\cdots$,$\ldots$, $\cdots$; (7) **g** represents the directions from each of n pair nodes $(\zeta_2, \psi_1^{\zeta_2})$, $(\zeta_2, \psi_2^{\zeta_2})$,$\cdots$, $(\zeta_2, \psi_m^{\zeta_2})$ to each of n pair nodes $(\zeta_n, \psi_1^{\zeta_n})$, $(\zeta_n, \psi_2^{\zeta_n})$,$\cdots$, $(\zeta_n, \psi_m^{\zeta_n})$; (8) **h** represents the directions from each of n pair nodes $\cdots$, $\cdots$,$\ldots$, $\cdots$ to each of n pair nodes $(\zeta_n, \psi_1^{\zeta_n})$, $(\zeta_n, \psi_2^{\zeta_n})$,$\cdots$, $(\zeta_n, \psi_m^{\zeta_n})$; (9) **i** represents the two-opposite-way directions from and to each of n pair nodes $\cdots$, $\cdots$,$\ldots$, $\cdots$ to and from each of n object nodes $x_1, x_2, \cdots, x_n$; and (10) **j** represents the directions from each of n pair nodes $(\zeta_n, \psi_1^{\zeta_n})$, $(\zeta_n, \psi_2^{\zeta_n})$,$\cdots$, $(\zeta_n, \psi_m^{\zeta_n})$ to each of n object nodes $x_1, x_2, \cdots, x_n$.

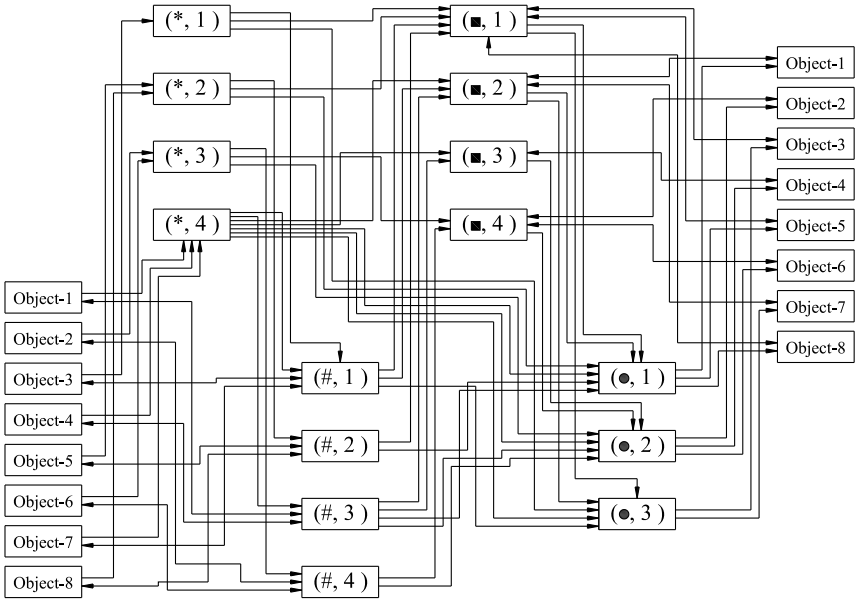


Fig. 15.8. Representation of a model DNA-digraph is used to reduct the given decision rules in the model decision table. In more detail, (1) in $(*,4)$, Object-1 is directed to $(\#,3)$, $(\blacksquare,2)$, and $(\bullet,1)$, Object-4 is directed to $(\#,3)$, $(\blacksquare,3)$, and $(\bullet,2)$, and Object-7 is directed to $(\#,1)$, $(\blacksquare,2)$, and $(\bullet,3)$; (2) in $(\#,1)$, Object-3 is directed to $(\blacksquare,1)$ and Object-7 is directed to $(\blacksquare,2)$; (3) in $(\#,3)$, Object-1 is directed to $(\blacksquare,2)$ and $(\bullet,1)$ and Object-4 is directed to $(\blacksquare,3)$ and $(\bullet,2)$; (4) in $(\blacksquare,1)$, Object-3 is directed to $(\bullet,3)$, Object-5 is directed to $(\bullet,1)$, and Object-8 is directed to $(\bullet,1)$; and (5) in $(\blacksquare,2)$, Object-1 is directed to $(\bullet,1)$ and Object-7 is directed to $(\bullet,3)$.

involves a directed relation to a pair node $(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})$, representing $x_i e(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})$, and $x_i\bar{e}(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})$ has no directed relation. The second case is when a pair node $(\zeta_\beta,\psi_\beta^{\zeta_\beta})$ involves a directed relation to an object node x_j, representing $(\zeta_\beta,\psi_\beta^{\zeta^\beta})ex_j$, and $(\zeta_\beta,\psi_\beta^{\zeta^\beta})\bar{e}x_j$ has no directed relation. The final case is when a pair node $(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})$ involves a directed relation to a pair node $(\zeta_\beta,\psi_\beta^{\zeta_\beta})$, representing $(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})e(\zeta_\beta,\psi_\beta^{\zeta_\beta})$, and $(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})\bar{e}(\zeta_\beta,\psi_\beta^{\zeta_\beta})$ has no directed relation.

In the DNA-digraph, three different types of related arcs among nodes exist for the representations of their arc relations. The first case describes a situation when an object node x_i involves a directed arrow, reaching a pair node $(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha})$, expressing

$$b_l^a = \overrightarrow{(x_i,(\zeta_\alpha,\psi_\alpha^{\zeta_\alpha}))}, i, \alpha, \text{ and } l = 1,2,\cdots,n. \tag{15.1}$$

The second case is when a pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ involves a directed arrow, reaching an object node x_j, expressing

$$b_l^b = \overrightarrow{((\zeta_\beta, \psi_\beta^{\zeta_\beta}), x_j)}, \beta, j, \text{ and } l = 1, 2, \cdots, n. \tag{15.2}$$

The final case is when a pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ involves a directed arrow, reaching another pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$, expressing

$$b_l^c = \overrightarrow{((\zeta_\alpha, \psi_\alpha^{\zeta_\alpha}), (\zeta_\beta, \psi_\beta^{\zeta_\beta}))}, \alpha, \beta, \text{ and } l = 1, 2, \cdots, n \text{ for } \alpha \neq \beta. \tag{15.3}$$

Three subsets of arcs are also represented in the following manner. The first case deals with a subset of arcs starting from the object node x_i and ending to the pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, denoted by B^a, consists of all of the arcs, representing $B^a = \{b_1^a, b_2^a, \cdots, b_n^a\}$. The second case involves a subset of arcs starting from the pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ and ending at the object node x_j, denoted by B^b, consists of all of the arcs, representing $B^b = \{b_1^b, b_2^b, \cdots, b_n^b\}$. The final case is when a subset of arcs starting from the pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ and ending at the pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$, denoted by B^c, consists of all of the arcs, representing $B^c = \{b_1^c, b_2^c, \cdots, b_n^c\}$.

The five different sets of U, P_s, B^a, B^b, and B^c are represented through a *binary adjacency matrix*. A model of a binary adjacency matrix can be constructed by applying the model DNA-digraph. It is composed of the rows and columns of the matrix, labeled as follows:

$$r_{i,j}, i \text{ and } j = 1, 2, \cdots, 23 \text{ for all } (i, j) \in L, \tag{15.4}$$

where a set of all the row and column labels is L, i is the row label, and j is the column label. When $r_{i,j} = 1$ in the model binary adjacency matrix, there are three arc relations: $x_i \rightarrow (\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, $(\zeta_\beta, \psi_\beta^{\zeta_\beta}) \rightarrow x_j$, and $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha}) \rightarrow (\zeta_\beta, \psi_\beta^{\zeta_\beta})$.

15.5.2 Encoding Process in DNA

To encode DNA sequences, the binary adjacency matrix is used in the encoding process. This type of matrix involves rows and columns, including the object and pair nodes, and is transformed into the desired DNA sequences.

Here, the model adjacency matrix is composed of 23 rows and 23 columns. In the model DNA-digraph, each of two directional nodes involves its own row and column. The 23 rows and 23 columns are labeled by $r_{i,j}$, i and $j = 1, 2, \cdots, 23$. For the DNA encoding process, we construct seven different types to create an initial DNA

library. Each of the seven types involves its own row and column labels. The seven types are referred to as types 1 to 7; types 1, 2, and 3 are substrings, which in this study are called *double-encoded substring*, and types 4, 5, 6, and 7 are complementary substrings. These double-encoded substrings and complementary substrings are all formed in ssDNA.

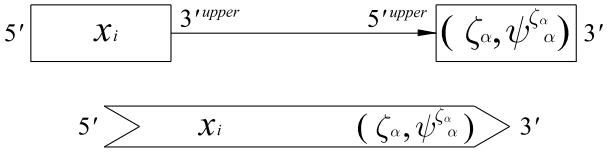


Fig. 15.9. Representation of a double-encoded substring for type 1

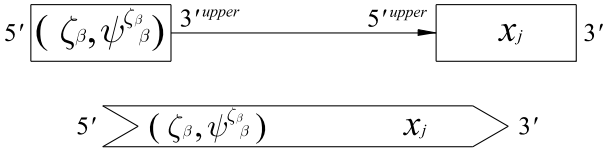


Fig. 15.10. Representation of a double-encoded substring for type 2

Type 1 expresses a double-encoded substring, shown in Fig. 15.9, which involves the two nodes x_i and $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, and contains a directed arrow, indicating the direction starting from x_i and ending at $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$. Thus, this type is denoted by a double-encoded substring $x_i - 3'^{upper} \rightarrow 5'^{upper} - (\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$. For Type 1, all the row and column labels are denoted by i and j, and those entries can be defined as

$$r_{i,j} = 1 \text{ for } i = 1, 2, \cdots, 8, j = 9, 10, \cdots, 20, \text{ and all } (i, j) \in L. \tag{15.5}$$

The two different nodes are encoded by a DNA oligonucleotide, in which two different sites have been unified into a single site. The single DNA oligonucleotide comes with its DNA base pair (bp) length (after hybridization and end-filling). Thus, for the model DNA-digraph, each object node $x_i, i = 1, 2, \cdots, 8$ was set to be a length of 18 bp, and each pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ represents $5'^{upper} - (* \vee \# \vee \blacksquare, 1 \vee 2 \vee 3 \vee 4)$, with a different bp length in each of the different pair nodes.

Type 2 also expresses a double-encoded substring, shown in Fig. 15.10, which involves the two nodes $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ and x_j, and contains a directed arrow that indicates the direction starting from $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ and ending at x_j. Thus, it is denoted by a double-encoded substring $(\zeta_\beta, \psi_\beta^{\zeta_\beta}) - 3'^{upper} \rightarrow 5'^{upper} - x_j$. For type 2, all the row and column labels are denoted as i and j, and those entries can be defined as

$$r_{i,j} = 1 \text{ for } i = 13, 14, \cdots, 23, j = 1, 2, \cdots, 8, \text{ and all } (i, j) \in L. \tag{15.6}$$

The two different nodes are encoded by a DNA oligonucleotide of a single site. The single DNA oligonucleotide involves its own DNA bp length. Thus, for the model DNA-digraph, each pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ represents $(\# \vee \blacksquare \vee \bullet, 1 \vee 2 \vee 3 \vee 4) - 3'^{upper}$, with a different bp length in each of the different pair nodes, and each object node x_j, $j = 1, 2, \cdots, 8$ was set to have a length of 18 bp.

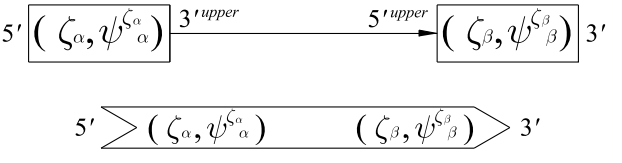


Fig. 15.11. Representation of a double-encoded substring for type 3.

Type 3 also expresses a double-encoded substring, shown in Fig. 15.11, which involves the two nodes $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ and $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$, and contains a directed arrow, indicating the direction starting from $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ and ending at $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$. Thus it is denoted by a double-encoded substring $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha}) - 3'^{upper} \rightarrow 5'^{upper} - (\zeta_\beta, \psi_\beta^{\zeta_\beta})$. For type 3, all the row and column labels are denoted as i and j, and those entries can be defined as

$$r_{i,j} = 1 \text{ for } i = 9, 10, \cdots, 20, j = 13, 14, \cdots, 23, \text{ and all } (i, j) \in L. \tag{15.7}$$

The two different pair nodes are encoded by a DNA oligonucleotide of a single site. The single DNA oligonucleotide involves its own DNA bp length. Thus, for the model DNA-digraph, each pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ represents $(* \vee \# \vee \blacksquare, 1 \vee 2 \vee 3 \vee 4) - 3'^{upper}$, with a different bp length in each of the different pair nodes, and each pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ represents $5'^{upper} - (\# \vee \blacksquare \vee \bullet, 1 \vee 2 \vee 3 \vee 4)$, with a different bp length in each of the different pair nodes.

In the DNA-digraph, the main forces of the hybridization and ligation techniques [4] with types 4, 5, 6, and 7 induce all three different DNA substrings to connect with one another. The complementary substrings of types 4, 5, 6, and 7 make the given double-encoded substrings connect clearly to become dsDNA.

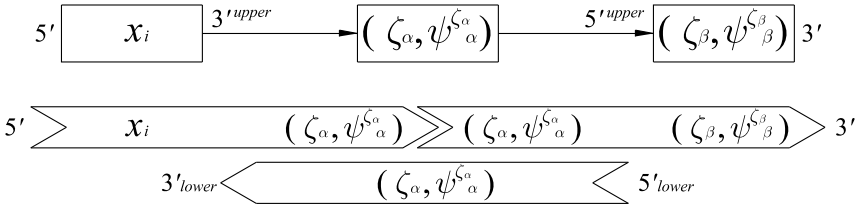


Fig. 15.12. Representation of a complementary substring for type 4

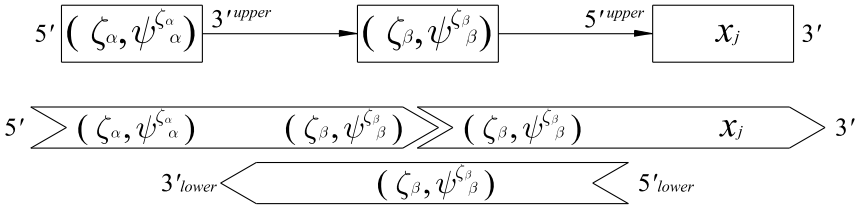


Fig. 15.13. Representation of a complementary substring for type 5

Type 4 makes an attachment of three different linked nodes, which correspond to an object node x_i, a pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ and a pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$, respectively, from $5'$ to $3'$, as shown in Fig. 15.12. A middle pair node of the complementary substring should be for type 4 generated to link together the three element nodes, and this complementary substring is denoted by a complementary encoding $5'_{lower} - (\zeta_\alpha, \psi_\alpha^{\zeta_\alpha}) \mid (\zeta_\alpha, \psi_\alpha^{\zeta_\alpha}) - 3'_{lower}$. For the model DNA-digraph, the middle pair node of the complementary substring represents $5'_{lower} - (1 \vee 2 \vee 3 \vee 4, * \vee \# \vee \blacksquare) \mid (1 \vee 2 \vee 3 \vee 4, * \vee \# \vee \blacksquare) - 3'_{lower}$.

Type 5 also realizes an attachment of three different linked nodes, which correspond to a pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, a pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ and an object node x_j, respectively, from $5'$ to $3'$, as shown in Fig. 15.13. Similar to type 4, a middle pair node

of the complementary substring should be generated with type 5. This complementary substring is denoted by a complementary substring encoding $5'_{lower} - (\zeta_\beta, \psi_\beta^{\zeta_\beta}) \mid (\zeta_\beta, \psi_\beta^{\zeta_\beta}) - 3'_{lower}$. The middle pair node of the complementary substring represents $5'_{lower} - (1 \vee 2 \vee 3 \vee 4, \# \vee \blacksquare \vee \bullet) \mid (1 \vee 2 \vee 3 \vee 4, \# \vee \blacksquare \vee \bullet) - 3'_{lower}$ for the middle DNA-digraph.

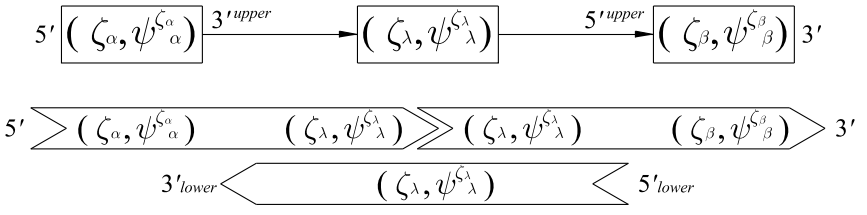


Fig. 15.14. Representation of a complementary substring for type 6

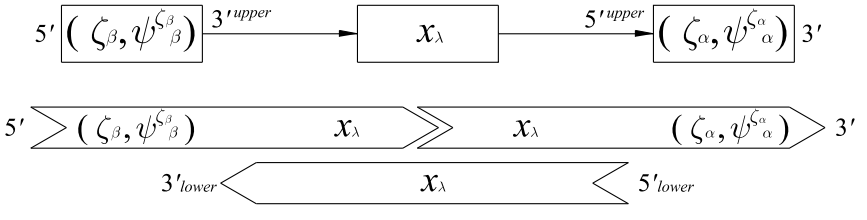


Fig. 15.15. Representation of a complementary substring for type 7

Type 6 also makes an attachment of three different linked nodes, which correspond to a pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, a pair node $(\zeta_\lambda, \psi_\lambda^{\zeta_\lambda})$ and a pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$, respectively, from $5'$ to $3'$, as shown in Fig. 15.14. Similarly to type 4, a middle pair node of the complementary substring should be generated with type 6. This complementary substring is denoted by a complementary substring encoding $5'_{lower} - (\zeta_\lambda, \psi_\lambda^{\zeta_\lambda}) \mid (\zeta_\lambda, \psi_\lambda^{\zeta_\lambda}) - 3'_{lower}$. The middle pair node of the complementary substring represents $5'_{lower} - (1 \vee 2 \vee 3 \vee 4, \# \vee \blacksquare) \mid (1 \vee 2 \vee 3 \vee 4, \# \vee \blacksquare) - 3'_{lower}$ for the model DNA-digraph.

Type 7 also makes an attachment of three different linked nodes, corresponding to a pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$, an object node x_λ and a pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, respectively,

from $5'$ to $3'$, as shown in Fig. 15.15. A middle object node of the complementary substring should be generated with type 7. This complementary substring is denoted by a complementary substring encoding $5'_{lower} - x_\lambda \mid x_\lambda - 3'_{lower}$. The middle object node o f the complementary substring represents $5'_{lower} - 1 \vee 2 \vee \cdots \vee 8 \mid 1 \vee 2 \vee \cdots \vee 8 - 3'_{lower}$ for the model DNA-digraph.

15.6 Experimental Studies

In what follows, we show the DNA rough-set computing method being applied to a splicing operational method [12] to form reducts of decision rules. The splicing operational method is executed by the proposed seven different types of double-encoded substrings and complementary substrings. For the simulated experiments in this study, the control program Vector NTI was employed to show the length-represented DNA strands, representing the final results of the decision rule reducts.

15.6.1 Experiments

The possible double-encoded substrings and complementary substrings among the object and pair nodes should be generated to first determine the lower approximation subsets in each of the decision classes and to secondly determine the decision rule reducts. The first process is executed by computing the DNA strand lengths, using substring types 3 and 6, second process is executed by determining all of the circular DNA fragments, using substring types 1, 2, 4, 5, and 7, as well as reusing types 3 and 6. Fig. 15.16 shows an example of the circular type of DNA fragment composed of Object-2, $(*,3)$, $(\blacksquare,4)$, and $(\bullet,2)$. In this example, the two pair nodes are linked together, and each of the pair nodes is also linked with the object node that is linked with the other pair node. The nodes now become the hybridized circular type of DNA fragment. In more detail, the protocol for DNA rough-set computing for decision rule reducts is described as follows:

Step 1

(DNA-digraph creation-1): By interpreting the given relations of each of the given objects, condition attributes, and condition attribute values, a DNA-digraph can be constructed with n object nodes and n pair nodes. As shown in Fig. 15.8, in the model DNA-digraph, each double-encoded substring of sequence directions is encoded based on the arc directions starting at an object node and ending at an object node.

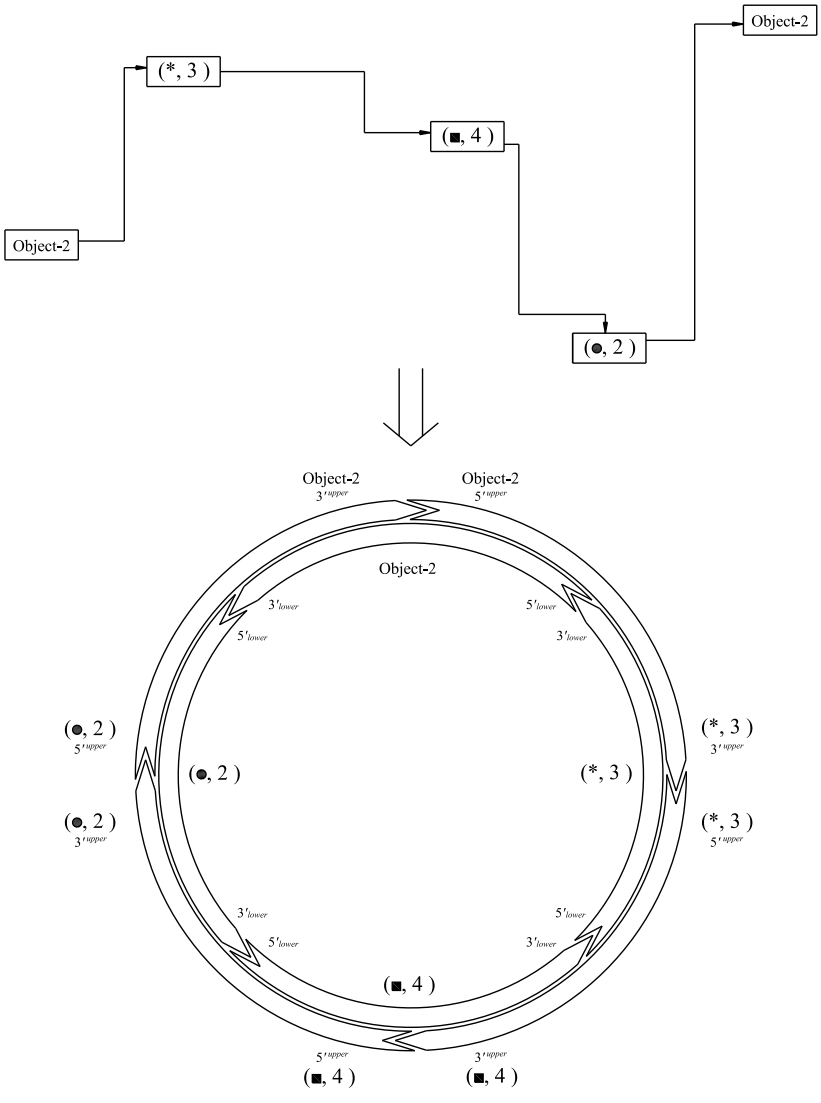


Fig. 15.16. Representation of an example direction composed of one object node and three different pair nodes becomes a circular DNA molecule from 5' to 3'

Step 2

(encoding process-1 and hybridization technique-1): The given object and pair nodes of the double-encoded substrings and their complementary substrings for the model DNA-digraph are encoded in DNA molecules. The type 3 and 6 substrings first generated and encoded in ssDNA are used initially to determine the lower approximation subsets in each decision class (we call this the *first-reduct*). All of the encoded type 3 double-encoded substrings and their type 6 complementary substrings are synthesized and placed in a prepared test tube. The encoded pair node of the double-encoded and complementary substrings for this hybridization technique based on the Watson-Crick complementary rules should be heated to about 94 °C and cooled to about 20 °C at multiple intervals of 1 °C/min.

Step 3

(simulated gel electrophoresis technique-1 and removal process-1): The lengths of all of the hybridized DNA strands are measured by the simulated gel electrophoresis technique, which can classify the strands into each of two decision classes (Decision Class-1 and -2) for the model decision table. If two or more hybridized DNA strands have the same length, then those strands correspond to the objects that involve the same condition attribute values. If two or more hybridized DNA strands with the same length are not involved in exactly the same decision class or are separately involved in each of the different decision classes despite having the same length, then those two or more hybridized DNA strands should be clearly removed from all decision classes. Subsequently, the remaining hybridized DNA strands for each decision class are divided into each lower approximation subset in each decision class.

Step 4

(denaturing process, marking process-1, and DNA-digraph creation-2): After the lower approximation subsets are determined in each decision class, types 3 and 6 of the hybridized DNA strands should be heated to about 94 °C to serve as ssDNA in the dsDNA status. All the denatured ssDNA strands are reused for the hybridization technique-2 and ligation technique. The number of DNA encoding sequences should be reducted by detecting one or more object nodes, which belong only to a subset of the lower approximation subsets and are involved in the exact same decision class. In more detail, if the two subsets, B_1 and B_2, of arcs between object and pair nodes are defined as

$$B_1 = \{\overrightarrow{(x_i, (\zeta_\alpha, \psi_\alpha^{\zeta_\alpha}))} \in B^a | x_i \in \Xi_A(D_\tau^\nu)\}, \tag{15.8}$$

which expresses a subset of arcs from an object node x_i to a pair node $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$, and

$$B_2 = \{(\overrightarrow{(\zeta_\beta, \psi_\beta^{\zeta_\beta}), x_j}) \in B^b | x_j \in \Xi_A(D_\tau^\nu)\}, \tag{15.9}$$

which expresses a subset of arcs from a pair node $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ to an object node x_j, then either x_i or x_j (or both x_i and x_j) in each subset of the object nodes is (are) involved in one subset of the lower approximation subsets. At the same time, each subset of pair nodes, either $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ or $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$ (or both $(\zeta_\alpha, \psi_\alpha^{\zeta_\alpha})$ and $(\zeta_\beta, \psi_\beta^{\zeta_\beta})$), is first marked as a first condition group of the reducted decision rules (we call this the *first part of second-reduct*). After determining the first part of second-reduct, an updated DNA-digraph should be created upon removing the first part of second-reduct, which corresponds to both B_1 and B_2. An updated model DNA-digraph is shown in Fig. 15.17.

Step 5

(encoding process-2, hybridization technique-2, ligation technique, restriction enzyme technique, and affinity separation technique-1): Based on the updated DNA-digraph, all of the double-encoded and complementary substrings can be encoded in ssDNA for types 1, 2, 4, 5, and 7, while reusing types 3 and 6. All of these types should also be synthesized and placed in the prepared test tube. For this step of the hybridization technique process, the ligases are added to bond the different encoded DNA sequences and to ensure their ligations. One or more circular DNA fragments should be detected and distinguished from all of the completely hybridized and ligated DNA strands. Here, the circular DNA fragments should be cut once at any point by employing the restriction enzyme technique to make them into linear DNA fragments, which should also be heated to about 94 °C to create ssDNA. The affinity separation technique-1 is also employed to classify each of the determined object groups (Object-1 to -8) in each object-labeled test tube.

Step 6

(simulated gel electrophoresis technique-2 and removal process-2): The simulated gel electrophoresis technique can measure the length of each of the object groups of the cut and denatured DNA strands. Each of the object groups of the DNA strands should be divided into the individual test tubes and should be loaded into different prepared lanes (object group lane-1 to -8). If there are two or more object groups of the DNA strands after all groups are loaded, indicating the presence of the same length, then those DNA strands should be clearly removed. In the same lower approximation subset, the removed DNA strands are obviously not involved.

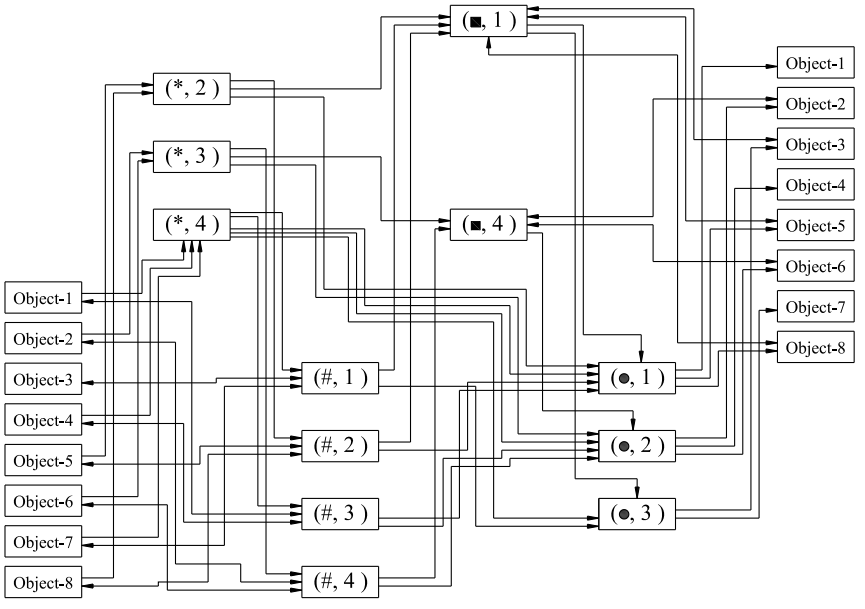


Fig. 15.17. Representation of an updated model DNA-digraph, in which the pair nodes of the first part of second-reduct have been removed.

Step 7

(affinity separation technique-2 and marking process-2): After removing DNA strands of the same length, each of the object group lanes (object group lane-1 to object group lane-8) should contain the remaining DNA fragments, which correspond to two or more pair nodes. Using the complementary substrings of the pair nodes, we distinguish all of the pair nodes contained in each of object group lanes (each of the low approximation subsets). Here, all of the pair nodes are the reducted decision rules (we call this the *second part of second-reduct*), which are divided into each decision class.

15.6.2 Experimental Results

The two techniques processing the simulated gel electrophoresis-1 and simulated gel electrophoresis-2 were executed to detect the DNA strands, representing the lower approximation subsets and decision rule reducts in each decision class. The results of experiments producing the lower approximation subsets and decision rule reducts in each decision class are shown in more detail below.

First (first-reduct), the two removable object groups, which were Object-2 and Object-6 (Decision Class-2 and Decision Class-1, respectively), and Object-5 and Object-8 (Decision Class-2 and Decision Class-1, respectively), had the same lengths. Hence, all of the remaining objects were Object-1 and Object-3 (Decision Class-2 and -1, respectively) or Object-4 and Object-7 (Decision Class-1 and -2, respectively). The lower approximation subsets in each of Decision Class-1 and Decision Class-2 were as follows: (1) Decision Class-1 = {Object-3, Object-4} derived from (Object-3) $(*,1)-3'^{upper} \rightarrow 5'^{upper}-(\#,1) \mid (\#,1)-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,1) \mid (\blacksquare,1)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,3)$ at 762 bp and (Object-4) $(*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,3) \mid (\#,3)-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,3) \mid (\blacksquare,3)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,2)$ at 1168 bp; and (2) Decision Class-2 = {Object-1, Object-7} derived from (Object-1) $(*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,3) \mid (\#,3)-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,2) \mid (\blacksquare,2)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,1)$ at 1128 bp and (Object-7) $(*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,1) \mid (\#,1)-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,2) \mid (\blacksquare,2)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,3)$ at 1032 bp.

Second (the first part of second-reduct), the results of the first marked condition group of the reducted decision rules in each decision class (Decision Class-1 and -2) were as follows: (1) Decision Class-1: (Object-3) $x_3-3'^{upper} \rightarrow 5'^{upper}-(*,1)$ and (Object-4) both $x_4-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,3)$ and $(\blacksquare,3)-3'^{upper} \rightarrow 5'^{upper}-x_4$; and (2) Decision Class-2: (Object-1) both $x_1-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,2)$ and $(\blacksquare,2)-3'^{upper} \rightarrow 5'^{upper}-x_1$ and (Object-7) both $x_7-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,2)$ and $(\blacksquare,2)-3'^{upper} \rightarrow 5'^{upper}-x_7$.

Finally (the second part of second-reduct), the results of the four different subsets of the pair nodes of the reducted decision rules in Object-1, Object-3, Object-4, and Object-7 in each decision class were as follows: (1) Decision Class-1: (Object-3) $x_3-3'^{upper} \rightarrow 5'^{upper}-(\#,1) \mid (\#,1)-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,1) \mid (\blacksquare,1)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,3) \mid (\bullet,3)-3'^{upper} \rightarrow 5'^{upper}-x_3$ at 568 bp, (Object-3) $x_3-3'^{upper} \rightarrow 5'^{upper}-(\#,1) \mid (\#,1)-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,1) \mid (\blacksquare,1)-3'^{upper} \rightarrow 5'^{upper}-x_3$ at 472 bp, (Object-3) $x_3-3'^{upper} \rightarrow 5'^{upper}-(\blacksquare,1) \mid (\blacksquare,1)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,3) \mid (\bullet,3)-3'^{upper} \rightarrow 5'^{upper}-x_3$ at 280 bp, (Object-4) $x_4-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,3) \mid (\#,3)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,2) \mid (\bullet,2)-3'^{upper} \rightarrow 5'^{upper}-x_4$ at 1556 bp, (Object-4) $x_4-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,2) \mid (\bullet,2)-3'^{upper} \rightarrow 5'^{upper}-x_4$ at 1148 bp, and (Object-4) $x_4-3'^{upper} \rightarrow 5'^{upper}-(\#,3) \mid (\#,3)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,2) \mid (\bullet,2)-3'^{upper} \rightarrow 5'^{upper}-x_4$ at 516 bp; and (2) Decision Class-2: (Object-1) $x_1-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,3) \mid (\#,3)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,1) \mid (\bullet,1)-3'^{upper} \rightarrow 5'^{upper}-x_1$ at 1532 bp, (Object-1) $x_1-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,1) \mid (\bullet,1)-3'^{upper} \rightarrow 5'^{upper}-x_1$ at 1124 bp, (Object-1) $x_1-3'^{upper} \rightarrow 5'^{upper}-(\#,3) \mid (\#,3)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,1)$—$(\bullet,1)-3'^{upper} \rightarrow 5'^{upper}-x_1$ at 492 bp, (Object-7) $x_7-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,1) \mid (\#,1)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,3) \mid (\bullet,3)-3'^{upper} \rightarrow 5'^{upper}-x_7$ at 1460 bp, (Object-7) $x_7-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\#,1) \mid (\#,1)-3'^{upper} \rightarrow 5'^{upper}-x_7$ at 1364 bp, and (Object-7) $x_7-3'^{upper} \rightarrow 5'^{upper}-(*,4) \mid (*,4)-3'^{upper} \rightarrow 5'^{upper}-(\bullet,3) \mid (\bullet,3)-3'^{upper} \rightarrow 5'^{upper}-x_7$ at 1172 bp.

The above results (first-reduct and the first and second parts of second-reduct) were interpreted as two different subsets of if-then rules, which were as follows: (1) Decision Class-1: if * is 1, then Decision Class-1 is chosen; if ■ is 3, then Decision Class-1 is chosen; if # is 1 and ■ is 1, then Decision Class-1 is chosen; if ■ is 1 and • is 3, then Decision Class-1 is chosen; if * is 4 and • is 2, then Decision Class-1 is chosen; and if # is 3 and • is 2, then Decision Class-1 is chosen; and (2) Decision Class-2: if ■ is 2, then Decision Class-2 is chosen; if * is 4 and • is 1, then Decision Class-2 is chosen; if # is 3 and • is 1, then Decision Class-2 is chosen; if * is 4 and # is 1, then Decision Class-2 is chosen; and if * is 4 and • is 3, then Decision Class-2 is chosen.

These interpreted results including the if-then decision rules in each decision class are the final solution of decision rule reducts obtained using DNA rough-set computing. It is worth stressing that the DNA rough-set computing method takes advantage of the characteristics of the various DNA molecular techniques.

To evaluate approximate computing times, we used the number of nodes (the number of object nodes and the number of pair nodes (composed of condition attributes and condition attribute values) in the DNA-digraph were set to become the number of nodes) and the number of directed arrows in the DNA-digraph to compare the proposed DNA rough-set computing method with an exponential-time computing method used for decision rule reducts. We selected the exponential-time computing method. To reach an optimal solution, the main reason was that the major problem was to find a way how we could reduct decision rules; recall that this problem is NP-hard, meaning that a polynomial-time computing method for this reduct problem has not been discovered yet.

In the exponential-time computing method, a processor executing 56,200 MIPS (mega instruction per second) (Intel Core2 Kentsfield Quad Core) was used, and the exponential-time computing method used an operation of 2^n [13]. The approximate computing times of the prepared DNA rough-set computing method were calculated based on the previously reported experimental results and genetic experimental notes [3]-[5]. When dealing with 60 nodes and 2,340 directed arrows, the prepared DNA rough-set computing method was approximately 2,307 times faster than the exponential-time computing one.

15.7 Conclusions

Based on the study of the proposed DNA rough-set computing for decision rule reduction, the three main conclusions can be presented.

First, this study shows a reduction of the complicated information of decision rules provided from the given decision using DNA rough-set computing. DNA rough-set computing was developed for the first-time in this study, and it offers some advantages and opens a new direction of optimization in our opinion. DNA rough-set computing was used to determine the lower approximation subsets

classified into each of the decision classes and to reduct all decision rules (or in other words, determine decision rules of minimal length).

Second, the study shows that DNA rough-set computing has a tangible potential for transforming a decision table into a digraph. This transformation created a DNA-digraph that can be applied not only to a reduct problem, but also to many others. DNA rough-set computing has also emerged as a potential way for combining different computer-based technologies from other fields to generate additional new computing methods.

Finally, the updating processes involving simplified data deliver more intuitive and useful information that can be readily handled and produced. However, the main difficulties still exist as we are faced with a number of intractable problems associated with such data. In this study, we showed that DNA rough-set computing can help to alleviate this kind of problems.

Acknowledgements. The authors would like to pay homage to Professor Zdzisław Pawlak and his scholarly accomplishments and achievements in numerous fields of innovative scientific and engineering studies.

This research was partly supported by the Kitakyushu Foundation for the Advancement of Industry, Science and Technology (FAIS) (FAIS R&D Funds for Scientific and Technical Collaborations among Asian Universities) in 2007-2011.

References

1. Adleman, L.M.: Computing with DNA: The manipulation of DNA to solve mathematical problems is redefining what is meant by computation. Scientific American 279(2), 34–41 (1998)
2. Burlage, R.S., Atlas, R., Stahl, D., Geesey, G., Sayler, G.: Techniques in microbial ecology, pp. 299–311. Oxford University Press Inc. (1998)
3. Burton, Z.F., Kaguni, J.M.: Experiments in molecularbiology: Biochemical applications, pp. 11–20. Academic Press (1997)
4. Cooper, N.G., Berg, P.: The human genome project: Deciphering the blueprint of heredity, pp. 48–54. University Science Books (1994)
5. Fitch, J.P.: An engineering introduction to biotechnology, pp. 43–60. SPIE Press (2002)
6. Goedken, E.R., Levitus, M., Johnson, A., Bustamante, C., O'Donnell, M., Kuriyan, J.: Fluorescence measurements on the E. coli DNA polymerase clamp loader: Implications for conformational changes during ATP and clamp binding. Journal of Molecular Biology 336(5), 1047–1059 (2004)
7. Greene, J.J., Rao, V.B.: Recombinant DNA principles and methodologies, pp. 126–134. Marcel Dekker Inc. (1996)
8. Grzymała-Busse, J.W.: A comparison of tree strategies to rule induction from data with numerical attributes. Electronic Notes in Theoretical Computer Science 82(4), 132–140 (2003)
9. Hartl, D.L., Jones, E.W.: Essential genetics: A genomics perspective, 3rd edn., pp. 90–412. Jones and Bartlett Publishers Inc. (2002)
10. Hartl, D.L., Jones, E.W.: Genetics: Analysis of genes and genomes, 7th edn., pp. 431–445. Jones and Bartlett Publishers Inc. (2009)

11. Jorgensen, W.L., Madura, J.D.: Temperature and size dependence for Monte Carlo simulation of TIP4P water. Molecular Physics: An International Journal at the Interface between Chemistry and Physics 56(6), 1381–1392 (1985)
12. Kari, L., Paun, G., Rozenberg, G., Salomaa, A., Yu, S.: DNA computing, sticker systems, and universality. Acta Informatica 35(5), 401–420 (1998)
13. Kleinberg, J., Tardos, E.: Algorithm design, pp. 29–207. Pearson Education Inc. (2006)
14. Matejtschuk, P.: Affinity separations: A practical approach, pp. 2–38. Oxford University Press Inc. (1997)
15. McPherson, M.J., Quirke, P., Taylor, G.R.: PCR 1: A practical approach, pp. 1–14. Oxford University Press Inc. (1991)
16. Micklos, D.A., Freyer, G.A., Crotty, D.A.: DNA science: A first course, 2nd edn., pp. 30–182. Cold Spring Harbor Laboratory Press (2003)
17. Nicholl, D.S.T.: An introduction to genetic engineering, 2nd edn. Studies in Biology, pp. 43–51. Cambridge University Press (2002)
18. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
19. Pawlak, Z.: Rough classification. International Journal of Man-Machine Studies 20(5), 469–483 (1984)
20. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: Probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29(1), 81–95 (1988)
21. Pedrycz, W., Chen, S.-C., Rubin, S.H., Lee, G.: Risk evaluation through decision-support architectures in threat assessment and countering terrorism. Applied Soft Computing 11(1), 621–631 (2011)
22. Pedrycz, W., Russo, B., Succi, G.: A model of job satisfaction for collaborative development processes. Journal of Systems and Software 84(5), 739–752 (2011)
23. Peters, J.F., Skowron, A.: Zdzisław Pawlak life and work (1926-2006). Information Sciences 177, 1–2 (2007)
24. Polkowski, L.: Advances in soft computing: Rough sets, pp. 18–45. Physica-Verlag, Heidelberg (2002)
25. Polkowski, L., Skowron, A.: Rough Sets in Knowledge Discovery 1. Methodology and Applications, pp. 201–250. Physica-Verlag, Heidelberg (1998)
26. Rabinow, P.: Making PCR: A story of biotechnology, pp. 1–17. The University of Chicago Press, Chicago (1996)
27. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.) Intelligent Decision Support: Handbook of Application and Advances of the Rough Sets Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
28. Shan, N., Ziarko, W.: Data-based acquisition and incremental modification of classification rules. Computational Intelligence 11(2), 357–370 (1995)
29. Skowron, A., Stepaniuk, J., Swiniarski, R.: Approximation spaces in rough-granular computing. Fundamenta Informaticae 100(1-4), 141–157 (2010)
30. Skowron, A., Wasilewski, P.: Information Systems in Modeling Interactive Computations on Granules. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS(LNAI), vol. 6086, pp. 730–739. Springer, Heidelberg (2010)
31. Słowiński, R. (ed.): Intelligent Decision Support. Handbook of Application and Advances of The Rough Sets Theory, pp. 363–372. Kluwer Academic Publishers, Dordrecht (1992)
32. Tyagi, R.: Understanding molecular biology, pp. 174–213. Discovery Publishing House Pvt. Ltd. (2009)

33. Watada, J., Wang, S., Pedrycz, W.: Building confidence-interval-based fuzzy random regression models. IEEE Transactions on Fuzzy System 17(6), 1273–1283 (2009)
34. Watson, J.D., Crick, F.H.C.: Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. Nature 171(4356), 737–738 (1953)
35. Watson, J.D., Myers, R.M., Caudy, A.A., Witkowski, J.A.: Recombinant DNA: Genes and genomes - a short course, 3rd edn., pp. 5–25. W.H. Freeman and Company (2007)
36. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)

Chapter 16
Three-Valued Logic for Reasoning about Covering-Based Rough Sets

Beata Konikowska

Abstract. In the chapter we present a tool for reasoning about covering-based rough sets in the form of three-valued logic in which the value **t** corresponds to the positive region of a set, the value **f** — to the negative region and the undefined value **u** — to the boundary of a given set. Atomic formulas of the logic represent either membership of objects of the universe in rough sets or the subordination relation between objects generated by the covering underlying the approximation space, and complex formulas are built out of the atomic ones using three-valued Kleene connectives. We give a strongly sound sequent calculus for the logic defined in this way and prove its strong completeness for a subset of its language.

Keywords: Rough sets, covering, subordination relation, rough set regions, three-valued logic, approximation space, Kleene connectives, sequent calculus.

16.1 Introduction

Rough sets, a very famous concept developed by Pawlak in the early 1980s [22, 23], are a both simple and powerful notion designed to model vague or imprecise information. In opposition to Zadeh's fuzzy sets, rough sets are not based on any numerical measure of the degree of membership of an object in an imprecisely defined set. Instead, they employ a much more universal and versatile idea of an indiscernibility relation, which groups together objects having the same properties from the viewpoint of a certain application into disjoint equivalence classes.

This concept has proved to be immensely useful in practice. Since their introduction in the early 1980s, rough sets have found numerous applications in areas like control of manufacturing processes [17], development of decisions tables [24], data

Beata Konikowska
Institute of Computer Science, Polish Academy of Sciences
e-mail: beatak@ipipan.waw.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 439–461.
springerlink.com

mining [17], data analysis [25], knowledge discovery [18, 19], and so on. They have also been the subject of an impressive body of research. Although the research focus has been mainly on the algebraic properties of rough sets, a number of logicians have also explored this area, presenting and studying various brands of rough set logics, including logics based on indiscernibility and similarity relations, as well as other types of relations considered in connection with imprecise information [8, 7, 10, 11, 15, 20, 21, 13, 12, 27, 28].

Over the years, one of the trends in rough set research involved generalizing that notion by replacing the indiscernibility relation underlying the original rough sets, which was an equivalence relation representing a partition of the universe of objects, with other, less restrictive constructs. The broadest generalization is covering-based rough sets [31, 26], whereby rough sets are defined based on an arbitrary covering of the universe of objects rather than on its partition. By now, this notion has also been examined in many papers (see *e.g.* [29, 30, 33]), with the main focus again on the algebraic properties of rough sets.

In opposition to the above, in this chapter we are interested in the logical aspects of rough sets. Our approach is based on employing three-valued logic. The underlying motivation is that, similarly to original Pawlak's rough sets, covering-based rough sets define three regions of any set X of objects, namely:

- positive region of X, containing all objects of the universe which *certainly belong* to X in the light of the information provided by the covering;
- negative region of X, containing all objects which *certainly do not belong* to X;
- boundary of X, containing all objects which cannot be said for sure to either belong or not to belong to X.

Hence, the most natural idea for reasoning about membership of objects in rough sets is to use a three-valued logic, with the values:

- **t** — meaning "certainly belongs" and assigned to objects in the positive region of a given set;
- **f** — meaning "certainly does not belong" and assigned to objects in the negative region of the set; and
- **u** — meaning "not known to either belong or not" and assigned to the boundary of the set.

Such an idea was exploited in [4] for the original rough sets based on an equivalence relation on the universe of objects. However, the logic developed there was just a simple propositional logic generated by the three-valued non-deterministic matrix (see [6, 2, 5, 1]), shortly: Nmatrix, $\mathcal{M}^3_{RS}$ with the following truth tables:

$-$	
f	**t**
u	**u**
t	**f**

$\cup$	**f**	**u**	**t**
f	**f**	**u**	**t**
u	**u**	$\{\mathbf{u},\mathbf{t}\}$	**t**
t	**t**	**t**	**t**

$\cap$	**f**	**u**	**t**
f	**f**	**f**	**f**
u	**f**	$\{\mathbf{f},\mathbf{u}\}$	**u**
t	**f**	**u**	**t**

(16.1)

where $\mathbf{f}, \mathbf{u}$ and $\mathbf{t}$ stand for the appropriate singleton sets. The reason for such an approach was that $\mathcal{M}^3_{RS}$ is observed by any interpretation I of formulas of the form $x \in A$ representing membership of an element x in a rough set A. More exactly, if $I(x\,\hat{\in}A) = a$ and $I(x\,\hat{\in}B) = b$, then $I(x\,\hat{\in}(A \diamond B))$ belongs to the set assigned to the pair (a,b) by the truth table of $\mathcal{M}^3_{RS}$ corresponding to the connective $\diamond$.

However, despite this fact, the above Nmatrix is not really adequate for describing set-theoretic operations on rough sets. Indeed: according to it, if $I(x\,\hat{\in}A) = \mathbf{u}$ and $I(x\,\hat{\in}B) = \mathbf{u}$, then $I(x\,\hat{\in}(A \cup B)) \in \{\mathbf{u}, \mathbf{t}\}$. The inclusion indeed holds — but having it as the only limitation on $I(x\,\hat{\in}(A \cup B))$ if $I(x\,\hat{\in}A) = I(x\,\hat{\in}B) = \mathbf{u}$ allows in case of $A = B$ for having $I(x\,\hat{\in}A) = \mathbf{u}$ and $I(x\,\hat{\in}(A \cup A)) = \mathbf{t}$ — which is inconsistent with the fact that $I_s(A \cup A) = I_s(A)$ under any normal interpretation I_s of set expressions. What is more, the above rule also allows for the case when $I(x\,\hat{\in}(A \cup B)) = \mathbf{u}$ and $I(x\,\hat{\in}(B \cup A)) = \mathbf{t}$, which is equally impossible. As a result, the sequent calculus developed in [4], after its lifting to the level of membership formulas, does not allow us to prove the sequents $x \in (A \cup A) \Rightarrow x \in A$ and $x \in (A \cup B) \Rightarrow x \in (B \cup A)$, which are valid in rough set models. Analogous inconsistencies occur in case of intersection.

To avoid such deficiencies, the semantics of the logic we will develop in this chapter will be based on natural frameworks for covering-based rough sets, that is, covering-based approximations spaces. Since the most interesting feature differing covering-based rough sets from the original, partition-based ones are the complex relations between objects generated by the usually overlapping elements of the covering, in our logic, we will explore a fundamental relation of that type, called *subordination relation*, and its relationships with the lower and upper approximations and the regions of a set, to better characterize those notions. Accordingly, the atomic formulas of the language of our logic will represent either membership of objects of the universe in rough sets or the subordination relation between objects. Complex formulas will be formed out of the atomic ones using negation, disjunction and conjunction, incorporated in the language in order to increase its expressiveness and interpreted according to the three-valued Kleene matrix. Our choice of such interpretation of the connectives was motivated by the wide use of Kleene connectives in applications related to computer software and computation. However, it should be noted that since the truth tables of Kleene and Łukasiewicz 3-valued matrices differ only in case of implication, which we do not use here, the interpretations of connectives employed in this chapter are also compliant with the semantics of Łukasiewicz three-valued logic [16], whose connections with rough set logics were studied in [8, 27].

The chapter is organized as follows. Section 16.2 presents the fundamentals of covering-based rough sets. In Section 16.3, the subordination relation is defined, and its properties are explored, including the closure properties of the lower and upper approximation and the regions of a set under that relation. Section 16.4 defines the syntax and semantics of the logic $\mathcal{L}^c_{RS}$ developed in the chapter, including satisfaction and consequence relations for formulas and sequents. Sections 16.5–16.8 present the proof system for $\mathcal{L}^c_{RS}$, and prove its strong soundness, as well as strong

completeness for the subset of the language featuring only atomic set expressions. Finally, Section 16.9 presents the conclusions and outlines future work.

16.2 Covering-Based Rough Sets

In what follows, for any set X, by $\mathcal{P}(X)$ we denote the powerset of X, that is, the set of all subsets of X and by $\mathcal{P}^+(X)$ — the set of all nonempty subsets of X.

Definition 16.1. By a *covering-based approximation space*, or shortly *approximation space*, we mean any ordered pair $\mathcal{A} = (U, \mathcal{C})$, where U is a non-empty universe of *objects*, and $\mathcal{C} \subseteq \mathcal{P}^+(U)$ is a covering of U, that is, $\bigcup\{C \mid C \in \mathcal{C}\} = \mathcal{U}$.

Definition 16.2. For any approximation space $\mathcal{A} = (U, \mathcal{C})$, we define the operators of *lower approximation* $L_\mathcal{C}$ and *upper approximation* $H_\mathcal{C}$ of subsets of U with respect to the covering $\mathcal{C}$ as follows:

- The *lower approximation* of a set $X \subseteq U$ with respect to the covering $\mathcal{C}$ is the set
$$L_\mathcal{C}(X) = \{x \in U \mid \forall C \in \mathcal{C}(x \in C \Rightarrow C \subseteq X)\}$$
- The *upper approximation* of a set $X \subseteq U$ with respect to the covering $\mathcal{C}$ is the set
$$H_\mathcal{C}(X) = \bigcup\{C \in \mathcal{C} \mid C \cap X \neq \emptyset\}$$

In view of the above definition, one can say that, given the approximate knowledge about objects available in the approximation space $\mathcal{A}$:

- $L_\mathcal{C}(X)$ is the set of all the objects in U which *certainly* belong to X;
- $H_\mathcal{C}(X)$ is the set of all the objects in U which *might* belong to X;

The above operations have the same basic properties as in case of Pawlak's rough sets based on a partition of the universe, that is, for any $X, Y \subseteq U$, we have

$$L_\mathcal{C}(X) \subseteq X \subseteq H_\mathcal{C}(X)$$

$$\begin{array}{ll} H_\mathcal{C}(X \cup Y) = H_\mathcal{C}X \cup H_\mathcal{C}Y & L_\mathcal{C}(X \cup Y) \supseteq L_\mathcal{C}X \cup L_\mathcal{C}Y \\ L_\mathcal{C}(X \cap Y) = L_\mathcal{C}X \cap L_\mathcal{C}Y & H_\mathcal{C}(X \cap Y) \subseteq H_\mathcal{C}X \cap H_\mathcal{C}Y \\ L_\mathcal{C}(-X) = -H_\mathcal{C}X & H_\mathcal{C}(-X) = -L_\mathcal{C}X \end{array} \tag{16.2}$$

where none of the inequalities in (16.2) can be replaced by an equality.

Following the example of Pawlak's rough sets, with any subset of a universe U of an approximation space, we can associate three regions of that universe: positive, negative, and boundary, representing three basic statuses of membership of an object of the universe U in X:

Definition 16.3. Let $\mathcal{A} = (U, \mathcal{C})$ be an approximation space, and let $X \subseteq U$. Then:

- The *positive region* of X in the space $\mathcal{A}$ with respect to the covering $\mathcal{C}$ is the set

$$POS_{\mathcal{C}}(X) = L_{\mathcal{C}}(X)$$

- The *negative region* of X with respect to the covering $\mathcal{C}$ is the set

$$NEG_{\mathcal{C}}(X) = L_{\mathcal{C}}(U - X)$$

- The *boundary region* of X with respect to the covering $\mathcal{C}$ is the set

$$BND_{\mathcal{C}}(X) = U - (POS_{\mathcal{C}}(X) \cup NEG_{\mathcal{C}}(X))$$

Corollary 16.1. *For any approximation space $\mathcal{A} = (U, \mathcal{C})$, and any $X \subseteq U$, we have:*

$$\begin{aligned} POS_{\mathcal{C}}(X) &= \{x \in U \mid \forall C \in \mathcal{C}(x \in C \Rightarrow C \subseteq X)\} \\ NEG_{\mathcal{C}}(X) &= \{x \in U \mid \forall C \in \mathcal{C}(x \in C \Rightarrow C \subseteq U - X)\} \\ BND_{\mathcal{C}}(X) &= \{x \in U \mid \exists C \in \mathcal{C}(x \in C \wedge C \cap X \neq \emptyset \wedge C \cap (U - X) \neq \emptyset\} \end{aligned} \tag{16.3}$$

The regions defined as above are obviously disjoint. Moreover, we can say that according to the approximate knowledge of the properties of objects in U provided by the covering $\mathcal{C}$:

- The elements of $POS_R(X)$ *certainly belong* to X;
- The elements of $NEG_R(X)$ *certainly do not belong* to X;
- *We cannot tell if* the elements of $BN_R(X)$ *belong* to X or not.

The above basic properties of regions imply that a natural logic for reasoning about covering-based rough sets is a three-valued logic with the values **t** — true, **f** – false, **u** — unknown, in which the value **t** corresponds to the positive region of a set, the value **f** — to the negative region, and the undefined value **u** — to the boundary region of a given set. This is exactly the approach we will adopt in defining our logic in what follows.

16.3 Subordination Relation and Closure Properties of Approximations and Regions

Assume $\mathcal{A} = (U, \mathcal{C})$ is an arbitrary approximation space to be fixed until stated to the contrary. In order to express the basic relationships between the objects in the universe U induced by the covering $\mathcal{C}$, we introduce a special *subordination relation*:

Definition 16.4. The relation of *subordination with respect to the covering* $\mathcal{C}$ is a binary relation $\prec_{\mathcal{C}}$ on U defined by:

$$x \prec_{\mathcal{C}} y \stackrel{df}{\equiv} \forall C \in \mathcal{C}(y \in C \Rightarrow x \in C)$$

Defining the notion of a *neighborhood* $(x)_{\mathcal{C}}$ *of an object* $x \in U$ *with respect to the covering* $\mathcal{C}$ as in the literature on covering-based rough sets, that is, by

$$(x)_{\mathcal{C}} = \bigcap\{C \in \mathcal{C} \mid x \in C\}$$

we immediately obtain the following connection between the two notions:

$$(\forall x, y \in U)[x \prec_{\mathcal{C}} y \Leftrightarrow x \in (y)_{\mathcal{C}}]$$

It can be easily seen that $\prec_{\mathcal{C}}$ is a partial quasi-order on U:

Lemma 16.1. *The subordination relation* $\prec_{\mathcal{C}}$ *is reflexive and transitive.*

Proof. By Def. 16.4, reflexivity is obvious, for $x \in C \rightarrow x \in C$. To show transitivity, assume (i) $x \prec_{\mathcal{C}} y$ and (ii) $y \prec_{\mathcal{C}} z$. Then, for any $C \in \mathcal{C}$, $z \in C$ implies $y \in C$ by (ii), whence $x \in C$ by (i). Thus $x \prec_{\mathcal{C}} z$, and $\prec_{\mathcal{C}}$ is indeed transitive.

By the above Lemma, the subordination relation is an equivalence relation iff it is symmetric. In particular, this is the case when $\mathcal{C}$ is a partition. Namely, we have:

Lemma 16.2. *If* $\mathcal{C}$ *is a partition, then* $\prec_{\mathcal{C}}$ *is the equivalence relation* $R_{\mathcal{C}}$ *on* U *corresponding to that partition, that is, the only equivalence relation with the set of equivalence classes identical with* $\mathcal{C}$.

Proof. If $\mathcal{C}$ is a partition, then different sets in $\mathcal{C}$ are disjoint. Hence, for any $y \in U$, there is a single $C \in \mathcal{C}$ such that $y \in C$. Consequently, $x \prec_{\mathcal{C}} y$ iff $x \in C$ iff $xR_{\mathcal{C}}y$.

In general, we have:

Corollary 16.2. *The relation* $\prec_{\mathcal{C}}$ *is an equivalence relation iff it is identical with the relation* $\sim_{\mathcal{C}}$ *defined by:*

$$x \sim_{\mathcal{C}} y \Leftrightarrow x \prec_{\mathcal{C}} y \wedge y \prec_{\mathcal{C}} x$$

Note that in view of the definition of $\prec_{\mathcal{C}}$, if the above condition holds, we have

$$(\forall x, y \in U)[(\forall C \in \mathcal{C})(y \in C \Rightarrow x \in C) \Leftrightarrow (\forall C \in \mathcal{C})(x \in C \Rightarrow y \in C)]$$

However, as the above does not imply that

$$(\forall x, y \in U)(\forall C \in \mathcal{C})[(y \in C \Rightarrow x \in C) \Leftrightarrow (x \in C \Rightarrow y \in C)]$$

then the converse to Lemma 16.2 does not hold — the covering $\mathcal{C}$ need not be a partition if $\prec_{\mathcal{C}}$ is an equivalence, because this relation does not uniquely determine the covering.

The subordination relation allows us to better characterize the lower and upper approximation and all the three regions of a subset of U. Namely, we have the following closure results:

Proposition 16.1. *For any subset X of U, the lower approximation (equivalently, the positive region of X) and the negative region of X with respect to $\mathcal{C}$ are upward-closed under $\prec_{\mathcal{C}}$, while the upper approximation of X and the boundary of X are downward-closed under $\prec_{\mathcal{C}}$. In other words, for any $x, y \in U$, the following implications hold:*

(i) *If $y \in L_{\mathcal{C}}(X)$ and $y \prec_{\mathcal{C}} x$, then $x \in L_{\mathcal{C}}(X)$*

(ii) *If $y \in H_{\mathcal{C}}(X)$ and $x \prec_{\mathcal{C}} y$, then $x \in H_{\mathcal{C}}(X)$*

(iii) *If $y \in Neg_{\mathcal{C}}(X)$ and $y \prec_{\mathcal{C}} x$, then $x \in Neg_{\mathcal{C}}(X)$*

(iv) *If $y \in Bnd_{\mathcal{C}}(X)$ and $x \prec_{\mathcal{C}} y$, then $x \in Bnd_{\mathcal{C}}(X)$.*

Proof. The mentioned implications are easily derived from Corollary 16.3 and Definitions 16.2, 16.3, 16.4, as shown below.

(i) Assume that $y \in L_{\mathcal{C}}(X)$ and $y \prec_{\mathcal{C}} x$, and consider any $C \in \mathcal{C}$. If $x \in C$, then $y \prec_{\mathcal{C}} x$ implies $y \in C$ by Definition 16.4. As $y \in L_{\mathcal{C}}(X)$, then, by Definition 16.2, $y \in C$ implies $C \subseteq X$. Since $x \in C$, in this way we have shown that, for any $C \in \mathcal{C}$, $x \in C$ implies $C \subseteq X$, whence $x \in L_{\mathcal{C}}(X)$ by Definition 16.2. Thus (i) holds.

(ii) Assume that $y \in H_{\mathcal{C}}(X)$ and $x \prec_{\mathcal{C}} y$. By Definition 16.2, $y \in H_{\mathcal{C}}(X)$ implies that there is $C \in \mathcal{C}$ such that $y \in C$ and $C \cap X \neq \emptyset$. Since $y \in C$ and $x \prec_{\mathcal{C}} y$, then by Definition 16.4 we have $x \in C$. Thus there is $C \in \mathcal{C}$ such that $x \in C$ and $C \cap X \neq \emptyset$, whence $x \in H_{\mathcal{C}}(X)$ by Definition 16.2. Hence (ii) holds.

(iii) Assume the left side of the implication holds. As $Neg_{\mathcal{C}}(X) = L_{\mathcal{C}}(U - X)$ by Definition 16.3, then we have $y \in L_{\mathcal{C}}(U - X)$ and $y \prec_{\mathcal{C}} x$. By the already proven implication (i) applied to $U - X$, this implies $x \in L_{\mathcal{C}}(U - X)$, whence $x \in Neg_{\mathcal{C}}(X)$, and (iii) holds.

(iv) Assume that $y \in Bnd_{\mathcal{C}}(X)$ and $x \prec_{\mathcal{C}} y$. By Definition 16.3, $y \in Bnd_{\mathcal{C}}(X)$ implies that there is $C \in \mathcal{C}$ such that $y \in C$ and both $C \cap X \neq \emptyset$ and $C \cap (U - X) \neq \emptyset$. Since $x \prec_{\mathcal{C}} y$, from $y \in C$ and Definition 16.4 we obtain $x \in C$. As a result, there is $C \in \mathcal{C}$ such that $x \in C$ and both $C \cap X \neq \emptyset$ and $C \cap (U - X) \neq \emptyset$. Hence $x \in Bnd_{\mathcal{C}}(X)$ by Definition 16.3 and (iv) holds too.

16.4 Syntax and Semantics of the Language L^c_{RS}

Now we shall define the language L^c_{RS} of the three-valued logic for reasoning about covering-based rough sets described in the introduction. Formulas of L^c_{RS} will contain expressions representing sets of objects (built from set variables using set-theoretic operators), variables representing objects, the symbol $\hat{\in}$ of a three-valued binary predicate representing membership of an object in a rough set, the symbol $\prec$ of a two-valued binary predicate representing the subordination relation on objects, and the logical connectives $\neg, \wedge, \vee$ which will be interpreted as 3-valued Kleene connectives.

Definition 16.5. Assume that:

- OV is a non-empty denumerable set of *object variables*;
- SV is a non-empty denumerable set of *set variables*.

The syntax of the language L^c_{RS} is defined as follows:

1. The set SE of *set expressions* of L^c_{RS} is the least set satisfying the following conditions:
   - $SV \subseteq SE$;
   - If $A, B \in SE$, then $-A, A \cup B, A \cap B \in SE$;
2. The set $\mathcal{A}^c_{RS}$ of *atomic formulas* of L^c_{RS} is the least set satisfying the following conditions:
   - If $x \in OV$ and $A \in SE$, then $x \hat{\in} A$ is in $\mathcal{A}^c_{RS}$;
   - For any $x, y \in OV$, the expression $x \prec y$ is in $\mathcal{A}^c_{RS}$.
3. The set $\mathcal{F}^c_{RS}$ of *formulas* of L^c_{RS} is the least set $\mathcal{F}$ containing $\mathcal{A}^c_{RS}$ and closed under the connectives $\neg, \vee, \wedge$.

The semantics of L^c_{RS} is based on interpreting the formulas of L^c_{RS} in semantic frameworks for that language built on covering-based approximation spaces and including valuations of set variables and object variables.

Definition 16.6. A *semantic framework*, or shortly *framework*, for L^c_{RS} is an ordered triple $\mathcal{R} = (\mathcal{A}, v, w)$, where

- $\mathcal{A} = (U, \mathcal{C})$ is a covering-based approximation space;
- $v : OV \to U$ is a valuation of object variables;
- $w : SV \to \mathcal{P}(U)$ is a valuation of set variables.

Definition 16.7. An *interpretation* of L^c_{RS} in a framework $\mathcal{R} = (\mathcal{A}, v, w)$, where $\mathcal{A} = (U, \mathcal{C})$, is a mapping $I_{\mathcal{R}} : \mathcal{F}^c_{RS} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ defined as follows:

1. $I_{\mathcal{R}}(x \prec y) = \begin{cases} \mathbf{t} \text{ if } v(x) \prec_{\mathcal{C}} v(y) \\ \mathbf{f} \text{ otherwise} \end{cases}$
2. $I_{\mathcal{R}}(x \hat{\in} A) = \begin{cases} \mathbf{t} \text{ if } v(x) \in Pos_{\mathcal{C}}(w^*(A)) \\ \mathbf{f} \text{ if } v(x) \in Neg_{\mathcal{C}}(w^*(A)) \\ \mathbf{u} \text{ if } v(x) \in Bnd_{\mathcal{C}}(w^*(A)) \end{cases}$

   for any $x, y \in OV$ and any $A \in SE$, where w^* is the extension of w to SE obtained by interpreting $-, \cup, \cap$ as the set-theoretic operations of complement, union and intersection, that is,
   - $w^*(X) = w(X)$ for any $X \in SV$,
   - for any $A, B \in SE$, $w^*(-A) = U - w(A)$, $w^*(A \cup B) = w^*(A) \cup w^*(B)$, $w^*(A \cap B) = w^*(A) \cap w^*(B)$.
3. For any $\varphi, \psi \in \mathcal{F}$:

$$\bullet\ I_{\mathcal{R}}(\neg\varphi) = \begin{cases} \mathbf{t} & \text{if } I_{\mathcal{R}}(\varphi) = \mathbf{f} \\ \mathbf{f} & \text{if } I_{\mathcal{R}}(\varphi) = \mathbf{t} \\ \mathbf{u} & \text{if } I_{\mathcal{R}}(\varphi) = \mathbf{u} \end{cases}$$

$$\bullet\ I_{\mathcal{R}}(\varphi \vee \psi) = \begin{cases} \mathbf{t} & \text{if either } I_{\mathcal{R}}(\varphi) = \mathbf{t} \text{ or } I_{\mathcal{R}}(\psi) = \mathbf{t} \\ \mathbf{f} & \text{if } I_{\mathcal{R}}(\varphi) = \mathbf{f} \text{ and } I_{\mathcal{R}}(\psi) = \mathbf{f} \\ \mathbf{u} & \text{otherwise} \end{cases}$$

$$\bullet\ I_{\mathcal{R}}(\varphi \wedge \psi) = \begin{cases} \mathbf{t} & \text{if } I_{\mathcal{R}}(\varphi) = \mathbf{t} \text{ and } I_{\mathcal{R}}(\psi) = \mathbf{t} \\ \mathbf{f} & \text{if either } I_{\mathcal{R}}(\varphi) = \mathbf{f} \text{ or } I_{\mathcal{R}}(\psi) = \mathbf{f} \\ \mathbf{u} & \text{otherwise} \end{cases}$$

It can be easily seen that the interpretation $I_{\mathcal{R}}$ is a well-defined mapping of the set of formulas into $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. Indeed, as the regions of a rough set are disjoint, Point 2 provides a well-defined interpretation of atomic membership formulas, and all the other clause raise no doubts. Note that the interpretation of subordination formulas given by Point 1 is two-valued.

In the future, we will drop the subscript at $I_{\mathcal{R}}$ whenever the framework $\mathcal{R}$ is understood.

16.4.1 Satisfaction and Consequence Relations for Formulas and Sequents

To complete the definition of the semantics of L^c_{RS}, we need to define the notion of satisfaction and the consequence relation. Since the proof system we are going to develop for L^c_{RS} will be a sequent calculus, we will define both the notions for formulas as well as for sequents.

Definition 16.8
By a *sequent*, we mean a structure of the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite *sets* of formulas. The set of all sequents in the language L^c_{RS} is denoted by Seq^c_{RS}.

We assume that the only designated value in the three-valued semantics of L^c_{RS} is $\mathbf{t}$. This gives rise to the following definitions of satisfaction and consequence:

Definition 16.9. 1. A formula $\varphi \in \mathcal{F}^c_{RS}$ is *satisfied* by an interpretation I of L^c_{RS}, in symbols $I \models \varphi$, if $v(\varphi) = \mathbf{t}$.
2. A formula $\varphi \in \mathcal{F}^c_{RS}$ is *valid*, in symbols $\models^c_{RS} \varphi$, if $I \models \varphi$ for any interpretation I of L^c_{RS}.
3. A set of formulas $T \subseteq \mathcal{F}^c_{RS}$ is *satisfied* by an interpretation I, in symbols $I \models T$, if $I \models \varphi$ for all $\varphi \in T$.
4. A sequent $\Sigma = (\Gamma \Rightarrow \Delta)$ is *satisfied* by an interpretation I, in symbols $I \models \Sigma$, iff either $I \models \varphi$ for some $\varphi \in \Delta$ or $I \not\models \psi$ for some $\psi \in \Gamma$.
5. A sequent $\Sigma = (\Gamma \Rightarrow \Delta)$ is *valid*, in symbols $\models^c_{RS} \Sigma$, if $I \models \Sigma$ for any interpretation I of L^c_{RS}

6. The *formula consequence relation* in L^c_{RS} is the relation $\vdash^c_{RS}$ on $\mathcal{P}(\mathcal{F}^c_{RS}) \times \mathcal{F}^c_{RS}$ such that, for every $T \subset \mathcal{F}^c_{RS}$ and every $\varphi \in \mathcal{F}^c_{RS}$

$$T \vdash^c_{RS} \varphi \text{ if every model of } T \text{ is also a model of } \varphi$$

7. The *sequent consequence relation* in L^c_{RS} is the relation $\vdash^c_{RS}$ on $\mathcal{P}(Seq^c_{RS}) \times Seq^c_{RS}$ such that, for every $Q \subseteq Seq^c_{RS}$, and every $\Sigma \in Seq^c_{RS}$

$$Q \vdash^c_{RS} \Sigma \text{ iff, for any interpretations } I \text{ of } L^c_{RS}, I \models^c_{RS} Q \text{ implies } I \models^c_{RS} \Sigma$$

16.5 Proof System for the Logic $\mathcal{L}^c_{RS}$

Now we shall present a proof system for the logic $\mathcal{L}^c_{RS}$ with the language L^c_{RS}, corresponding to the consequence relation $\vdash^c_{RS}$ defined in the preceding chapter. It can be easily noted that, in view of the properties of Kleene connectives, the only tautologies in this logic express the properties of the two-valued subordination relation rather than the properties of rough sets themselves. Accordingly, our proof system should be aimed at proving entailments, which are represented by sequents. As a result, the deduction formalism we will use for $\mathcal{L}^c_{RS}$ is a sequent calculus.

16.6 Sequent Calculus CRS

Let CRS be the sequent calculus over $\mathcal{L}^c_{RS}$ defined as follows:

Axioms:

$$(A1)\ \varphi \Rightarrow \varphi \qquad (A2)\ \neg\varphi, \varphi \Rightarrow$$

$$(A3)\ z \prec x, z \prec y, x \hat{\in} A, y \hat{\in} -A \Rightarrow \qquad (A4)\ \Rightarrow x \hat{\in} (A \cup -A)$$

Structural rules: weakening, cut

Inference rules for set-theoretic operations:

$$(--\Rightarrow)\ \frac{\Gamma, x \hat{\in} A \Rightarrow \Delta}{\Gamma, x \hat{\in} --A \Rightarrow \Delta} \qquad (\Rightarrow --)\ \frac{\Gamma \Rightarrow \Delta, x \hat{\in} A}{\Gamma \Rightarrow \Delta, x \hat{\in} --A}$$

$$(id \cup \Rightarrow)\ \frac{\Gamma, x \hat{\in} A \Rightarrow \Delta}{\Gamma, x \hat{\in} A \cup A \Rightarrow \Delta}$$

$$(\cup\Rightarrow)_a \quad \frac{\Gamma\Rightarrow\Delta, x\,\hat{\in}-A \quad \Gamma, x\,\hat{\in}B\Rightarrow\Delta}{\Gamma, x\,\hat{\in}A\cup B\Rightarrow\Delta} \qquad (\Rightarrow\cup) \quad \frac{\Gamma,\Rightarrow\Delta, x\,\hat{\in}A, x\,\hat{\in}B}{\Gamma\Rightarrow\Delta, x\,\hat{\in}A\cup B}$$

$$(\cup\Rightarrow)_b \quad \frac{\Gamma\Rightarrow\Delta, x\,\hat{\in}-B \quad \Gamma, x\,\hat{\in}A\Rightarrow\Delta}{\Gamma, x\,\hat{\in}A\cup B\Rightarrow\Delta}$$

$$(-\cup\Rightarrow) \quad \frac{\Gamma, x\,\hat{\in}-A, x\,\hat{\in}-B\Rightarrow\Delta}{\Gamma, x\,\hat{\in}-(A\cup B)\Rightarrow\Delta} \qquad (\Rightarrow-\cup) \quad \frac{\Gamma\Rightarrow\Delta, x\,\hat{\in}-A \quad \Gamma\Rightarrow\Delta, x\,\hat{\in}-B}{\Gamma\Rightarrow\Delta, x\,\hat{\in}-(A\cup B)}$$

$$(\cap\Rightarrow) \quad \frac{\Gamma, x\,\hat{\in}A, x\,\hat{\in}B\Rightarrow\Delta}{\Gamma, x\,\hat{\in}A\cap B\Rightarrow\Delta} \qquad (\Rightarrow\cap) \quad \frac{\Gamma\Rightarrow\Delta, x\,\hat{\in}A \quad \Gamma\Rightarrow\Delta, x\in B}{\Gamma\Rightarrow\Delta, x\,\hat{\in}A\cap B}$$

$$(id\ -\cap\Rightarrow) \quad \frac{\Gamma, x\,\hat{\in}-A\Rightarrow\Delta}{\Gamma, x\,\hat{\in}-(A\cap A)\Rightarrow\Delta}$$

$$(-\cap\Rightarrow)_a \quad \frac{\Gamma, x\,\hat{\in}-A\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta, x\,\hat{\in}B}{\Gamma, x\,\hat{\in}-(A\cap B)\Rightarrow\Delta} \qquad (\Rightarrow-\cap) \quad \frac{\Gamma\Rightarrow\Delta, x\,\hat{\in}-B, x\,\hat{\in}-A}{\Gamma\Rightarrow\Delta, x\,\hat{\in}-(A\cap B)}$$

$$(-\cap\Rightarrow)_b \quad \frac{\Gamma, x\,\hat{\in}-B\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta, x\,\hat{\in}A}{\Gamma, x\,\hat{\in}-(A\cap B)\Rightarrow\Delta}$$

Inference rules for Kleene negation over atomic formulas:

$$(\neg\prec\Rightarrow) \quad \frac{\Gamma\Rightarrow\Delta, x\prec y}{\Gamma, \neg(x\prec y)\Rightarrow\Delta} \qquad (\Rightarrow\neg\prec) \quad \frac{\Gamma, x\prec y\Rightarrow\Delta}{\Gamma\Rightarrow\Delta, \neg(x\prec y)}$$

$$(\neg\,\hat{\in}\Rightarrow) \quad \frac{\Gamma, x\,\hat{\in}-A\Rightarrow\Delta}{\Gamma, \neg(x\,\hat{\in}A)\Rightarrow\Delta} \qquad (\Rightarrow\neg\,\hat{\in}) \quad \frac{\Gamma\Rightarrow\Delta, x\,\hat{\in}-A}{\Gamma\Rightarrow\Delta, \neg(x\,\hat{\in}A)}$$

Inference rules for Kleene connectives:

$$(\neg\neg\Rightarrow) \quad \frac{\Gamma, \varphi\Rightarrow\Delta}{\Gamma, \neg\neg\varphi\Rightarrow\Delta} \qquad (\Rightarrow\neg\neg) \quad \frac{\Gamma\Rightarrow\Delta, \varphi}{\Gamma\Rightarrow\Delta, \neg\neg\varphi}$$

$$(\vee\Rightarrow) \quad \frac{\Gamma, \varphi\Rightarrow\Delta \quad \Gamma, \psi\Rightarrow\Delta}{\Gamma, \varphi\vee\psi\Rightarrow\Delta} \qquad (\Rightarrow\vee) \quad \frac{\Gamma,\Rightarrow\Delta, \varphi, \psi}{\Gamma\Rightarrow\Delta, \varphi\vee\psi}$$

$$(\neg\vee\Rightarrow) \quad \frac{\Gamma, \neg\varphi, \neg\psi\Rightarrow\Delta}{\Gamma, \neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad (\Rightarrow\neg\vee) \quad \frac{\Gamma\Rightarrow\Delta, \neg\varphi \quad \Gamma\Rightarrow\Delta, \neg\psi}{\Gamma\Rightarrow\Delta, \neg(\varphi\vee\psi)}$$

$$(\wedge\Rightarrow) \quad \frac{\Gamma, \varphi, \psi\Rightarrow\Delta}{\Gamma, \varphi\wedge\psi\Rightarrow\Delta} \qquad (\Rightarrow\wedge) \quad \frac{\Gamma\Rightarrow\Delta, \varphi \quad \Gamma\Rightarrow\Delta, \psi}{\Gamma\Rightarrow\Delta, \varphi\wedge\psi}$$

$$(\neg\wedge\Rightarrow) \quad \frac{\Gamma, \neg\varphi\Rightarrow\Delta \quad \Gamma, \neg\psi\Rightarrow\Delta}{\Gamma, \neg(\varphi\wedge\psi)\Rightarrow\Delta} \qquad (\Rightarrow\neg\wedge) \quad \frac{\Gamma\Rightarrow\Delta, \neg\varphi, \neg\psi}{\Gamma\Rightarrow\Delta, \neg(\varphi\wedge\psi)}$$

Analytic omission (closure) rules:

$$(refl\ \prec)\ \frac{\Gamma, x \prec x \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad \text{where } x \text{ occurs in } \Gamma \cup \Delta$$

$$(tr\ \prec)\quad \frac{\Gamma, x \prec y, y \prec z, x \prec z \Rightarrow \Delta}{\Gamma, x \prec y, y \prec z \Rightarrow \Delta} \qquad (cl\ \hat{\in}\ \prec)\ \frac{\Gamma, y \prec x, y\ \hat{\in} A, x\ \hat{\in} A \Rightarrow \Delta}{\Gamma, y \prec x, y\ \hat{\in} A \Rightarrow \Delta}$$

$$(sym\ \cup)\qquad \frac{\Gamma, x \in (A \cup B), x \in (B \cup A) \Rightarrow \Delta}{\Gamma, x \in (A \cup B) \Rightarrow \Delta}$$

$$(sym\ -\cap)\qquad \frac{\Gamma, x \in -(A \cap B), x \in -(B \cap A) \Rightarrow \Delta}{\Gamma, x \in -(A \cap B) \Rightarrow \Delta}$$

In all axioms and inference rules, we assume that $x, y, z \in OV$ and $A, B \in SE$.

It should be noted that the axioms and inference rules for set-theoretic operations are those given in [4] lifted to the level of membership formulas, supplemented by axioms A3 and A4, as well the two "idempotence rules" $(id \cup \Rightarrow), (id - \cap \Rightarrow)$, and the two "symmetry rules" $(sym\ \cup), (sym\ -\cap)$, which have been added to remedy the deficiencies of the system given in [4] mentioned in the introduction.

Besides the axioms given above, in what follows, we will sometimes use the derived axiom

$$(A5)\ x\ \hat{\in} A, x\ \hat{\in} -A \Rightarrow \tag{16.4}$$

To derive (A5), we take $x = y = z$ in Axiom A3, which yields $x \prec x, x\ \hat{\in} A, x\ \hat{\in} -A \Rightarrow$, and then eliminate $x \prec x$ by applying rule (refl $\prec$).

16.7 Soundness of CRS

Lemma 16.3

1. *The axioms of the system CRS are valid.*
2. *For any inference rule* ρ *of CRS and any framework* $\mathcal{R}$ *for* L^c_{RS}*, if the interpretation* I *of* L^c_{RS} *in* $\mathcal{R}$ *satisfies all the premises of* ρ*, then* I *satisfies the conclusion of* ρ *as well.*

Proof. Both parts can be easily verified based on the individual clauses of the definition of I given in Definition 16.7.

Since the rules for Kleene connectives are well known, by way of example, we shall prove the validity of the rather non-obvious Axiom A3, the omission rule $(cl\ \hat{\in}\ \prec)$, and one of the very non-standard rules for introducing $\cup$ and $-\cap$ on the left hand side of the sequent, namely rule $(\cup \Rightarrow)_a$.

Assume I is an interpretation of $\mathcal{L}^c_{RS}$ in a framework $\mathcal{R} = \{(U, \mathcal{C}), v, w\}$.

We begin with axiom A3 having the form $z \prec x, z \prec y, x\,\hat{\in} A, y\,\hat{\in} -A \Rightarrow$. Suppose $I \models \{z \prec x, z \prec y, x\,\hat{\in} A, y\,\hat{\in} -A\}$. Then, by Definition 16.7, $z \prec_C x, z \prec_C y, v(x) \in Pos(w^*(A)), v(y) \in Pos(w^*(-A))$. Since by Proposition 16.1 both $Pos(w^*(A)$ and $Pos(w^*(-A)$ are closed under the subordination relation $\prec_C$, this implies $v(z) \in Pos(w^*(A))$ and $v(z) \in Pos(w^*(-A))$ — which is a contradiction, because by (16.3) $Pos(w^*(-A)) = U - Pos(w^*(A))$, so these two sets are disjoint. As $\mathcal{R}$ is an arbitrary framework, this shows that the left-hand side of A3 is unsatisfiable, whence the axiom is valid.

Now let us pass to the rule

$$(cl\ \hat{\in} \prec) \qquad \frac{\Gamma, y \prec x, y\,\hat{\in} A, x\,\hat{\in} A \Rightarrow \Delta}{\Gamma, y \prec x, y\,\hat{\in} A \Rightarrow \Delta}$$

and assume (*) $I \models (\Gamma, y \prec x, y\,\hat{\in} A, x\,\hat{\in} A \Rightarrow \Delta)$. Suppose now $I \models \{\Gamma, y \prec x, y\,\hat{\in} A\}$. Then, $v(y) \prec_C v(x)$ and $v(y) \in Pos(w^*(A))$, so by Proposition 16.1, we have $v(x) \in Pos(w^*(A))$. Hence, $I \models \{\Gamma, y \prec x, y\,\hat{\in} A, x\,\hat{\in} A\}$, which yields $I \models \Delta$ by (*). As a result, $I \models (\Gamma, y \prec x, y\,\hat{\in} A \Rightarrow \Delta)$, which proves the soundness of the rule.

Consider finally the rule

$$(\cup \Rightarrow)_a \qquad \frac{\Gamma \Rightarrow \Delta, x\,\hat{\in} -A \quad \Gamma, x\,\hat{\in} B \Rightarrow \Delta}{\Gamma, x\,\hat{\in} A \cup B \Rightarrow \Delta}$$

Denote the premises by S_1, S_2, and assume $I \models S_1, S_2$. Suppose now $I \models \Gamma, x\,\hat{\in} A \cup B$. Then, $v(x) \in Pos(w^*(A \cup B))$, which means that, for every $C \in \mathcal{C}$, $x \in C$ implies $C \subseteq w^*(A \cup B)$. As $w^*(A \cup B) = w^*(A) \cup w^*(B)$, this implies that $C \subseteq w^*(A)$ or $C \subseteq w^*(B)$ or C has non-empty intersections with both the latter sets and their complements. Hence, we have the following three possibilities:

$v(x) \in Pos(w^*(A))$ Then, $I \not\models (x\,\hat{\in} -A)$, which in view of $I \models S_1$ yields $I \models \Delta$;
$v(x) \in Pos(w^*(B))$ Then, $I \models (x\,\hat{\in} B)$, which yields $I \models \Delta$ by the fact that $I \models S_2$;
$v(x) \in Bnd(w^*(A)) \cap Bnd(w^*(B))$ Then, once more $I \not\models (x\,\hat{\in} -A)$, whence $I \models \Delta$ by the fact that $I \models S_1$.

Hence, this rule is also sound.

Clearly, from the above Lemma, we can immediately conclude that:

Corollary 16.3. *The inference rules of CRS are strongly sound, that is, they preserve the validity of sequents.*

16.8 Completeness of a Sublanguage

Unfortunately, because of the technicalities resulting from the complex character of covering-based approximation spaces underlying the semantics of $\mathcal{L}^c_{RS}$, for the time being there is no proof that *CRS* is complete for the full language L^c_{RS}.

However, we will prove the completeness of CRS — or rather its appropriate subsystem — for the subset $\mathcal{F}^{at}_{RS}$ of the language $\mathcal{L}^{c}_{RS}$ consisting of formulas which do not contain complex set expressions. The importance of this language follows from the fact that it contains both formulas representing membership of elements in rough sets and formulas representing the subordination relation between objects generated by the covering, and exploiting the relationship between the latter relation and membership of elements in rough sets allows us to get better insight in the way in which a given covering generates rough sets.

In what follows, Seq^{at}_{RS} denotes the set of all sequents over formulas in $\mathcal{F}^{at}_{RS}$, and $\vdash^{at}_{RS}, \vdash^{\not\prec}_{RS}$ — the consequence relation $\vdash^{c}_{RS}$ restricted to sets of formulas and formulas in $\mathcal{F}^{at}_{RS}$.

Theorem 16.1. *Let CRS^{at} be the calculus obtained from CRS by:*

- *deleting all inference rules for set-theoretic operations, together with the rules* $(\neg\, \hat{\in} \Rightarrow), (\Rightarrow \neg\, \hat{\in}), (sym\ \cup), (sym\ -\cap)$ *and Axiom A4;*
- *replacing Axiom A3 with*

$$(A3') \qquad z \prec x, z \prec y, x\ \hat{\in} A, \neg(y\ \hat{\in} A) \Rightarrow$$

- *adding the rule*

$$(cl\ \neg\, \hat{\in} \prec) \qquad \frac{\Gamma, y \prec x, \neg(y\ \hat{\in} A), \neg(x\ \hat{\in} A) \Rightarrow \Delta}{\Gamma, y \prec x, \neg(y\ \hat{\in} A) \Rightarrow \Delta}$$

Then the calculus CRS^{at} is finitely strongly sound and complete for $\vdash^{at}_{RS}$, that is, for any finite set of sequents $S \subseteq Seq^{at}_{RS}$ and any sequent $\Sigma \in Seq^{at}_{RS}$, $S \vdash^{at}_{RS} \Sigma$ iff $S \vdash_{CRS^{at}} \Sigma$.

Proof. The soundness part follows from the soundness of CRS proven in Lemma 16.3, together with the fact that Axiom $A3'$ is derivable in CRS from Axiom $A3$ and rule $(\neg \prec \Rightarrow)$, while rule $(cl\ \neg\, \hat{\in} \prec)$ is derivable from rules $(\neg \prec \Rightarrow), (cl\ \hat{\in} \prec)$.

To prove the completeness part, we argue by contradiction. Suppose that for a finite set of sequents S and a sequent $\Sigma_0 = (\Gamma \Rightarrow \Delta)$, we have $S \vdash^{at}_{RS} \Sigma_0$, but Σ_0 is not derivable from S in CRS^{at}. We shall construct a counter-model for the entailment $S \vdash^{at}_{RS} \Sigma_0$, that is, a framework $\mathcal{R}$ for L^{c}_{RS} such that $I_{\mathcal{R}} \models S$ but $I_{\mathcal{R}} \not\models \Sigma_0$. The general method employed in the proof is based on the use of saturated sequents and is similar to that employed in [3, 4],

Denote by $F(S)$ the set of all formulae belonging to at least one side of some sequent in S. Then $F(S)$ is finite; assume it has l elements. Let $\varphi_1, \varphi_2, \ldots, \varphi_l$ be an enumeration of the formulae in $F(S)$. We shall now define a sequence of sequents $\Gamma_n \Rightarrow \Delta_n, n = 0, 1, \ldots, l$, such that, for $n = 0, 1, \ldots, l$:

(i) $\Gamma \subseteq \Gamma_n$, $\Delta \subseteq \Delta_n$
(ii) If $n \neq 0$, then $\varphi_n \in (\Gamma_n \cup \Delta_n)$.
(iii) $\Gamma_n \Rightarrow \Delta_n$ is not derivable from S in CRS^{at}.

The above sequence is defined inductively as follows:

- We put $\Gamma_0 = \Gamma, \Delta_0 = \Delta$. As by our assumption $\Gamma \Rightarrow \Delta$ is not derivable from S in CRS^{at}, the conditions (i)–(iii) above are satisfied for $n = 0$.

- Suppose $n \leq l-1$ and we have defined the sequents $\Gamma_i \Rightarrow \Delta_i$ satisfying conditions (i)–(iii) for $i \leq n$. Then, the sequents $\Sigma_1 = \Gamma_n \Rightarrow \Delta_n, \varphi_{n+1}$ and $\Sigma_2 = \varphi_{n+1}, \Gamma_n \Rightarrow \Delta_n$ cannot be both derivable from S in CRS^{at}, since then $\Gamma_n \Rightarrow \Delta_n$ would be derivable from them by cut on the formula φ_{n+1}. We take $\Gamma_{n+1} \Rightarrow \Delta_{n+1}$ to be Σ_1, if Σ_1 is not derivable from S, and Σ_2 otherwise. Then, obviously, from the inductive assumption it follows that the sequence $\Gamma_{n+1} \Rightarrow \Delta_{n+1}$ satisfies conditions (i)–(iii).

By induction, the whole sequence $\Gamma_n \Rightarrow \Delta_n, n = 0, 1, \ldots, l$, satisfies the desired conditions (i)–(iii). What is more, from the inductive construction we can see that

(iv) $\Gamma_n \subseteq \Gamma_{n+1}, \Delta_n \subseteq \Delta_{n+1}$ for $n = 1, 2, \ldots, l-1$

Let $\Sigma^* = (\Gamma^* \Rightarrow \Delta^*)$ be the extension of $\Gamma_l \Rightarrow \Delta_l$ to a saturated sequent, that is, a minimal sequent containing $\Gamma_l \Rightarrow \Delta_l$ and closed under the logical rules in CRS^{at} applied backwards. By way of example, a sequent $\Gamma' \Rightarrow \Delta'$ is closed under rule $(\vee \Rightarrow)$ applied backwards if whenever $\varphi \vee \psi$ is in Γ', either φ is in Γ' or ψ is in Γ', and it is closed under rule $(\Rightarrow \vee)$ applied backwards if whenever $\varphi \vee \psi$ is in Δ', both φ and ψ are in Δ'.

Then, we can easily see that

(I) $\Gamma \subseteq \Gamma^*, \Delta \subseteq \Delta^*$

(II) $F(S) \subseteq \Gamma^* \cup \Delta^*$

(III) $\Gamma^* \Rightarrow \Delta^*$ is saturated, and it is not derivable from S in CRS^{at}.

Now we construct the desired counter-model $\mathcal{R} = (\mathcal{A}, v, w)$. Define

$$OV_0 = \{x \in OV \mid x \text{ occurs in } \Sigma^*\}, \qquad SV_0 = \{S \in SV \mid S \text{ occurs in } \Sigma^*\} \quad (16.5)$$

(recall that formulas in $\mathcal{F}_{RS}^{at}$ do not contain any complex set expressions).

Let R_t, R_f, R_u be the relations on $OV_0 \times SV_0$ defined by:

$$\begin{gathered} R_t(x,A) \Leftrightarrow (x\,\hat{\in} A) \in \Gamma^* \qquad R_f(x,A) \Leftrightarrow \neg(x\,\hat{\in} A) \in \Gamma^* \\ R_u(x,A) \Leftrightarrow \neg R_t(x,A) \wedge \neg R_f(x,A) \end{gathered} \quad (16.6)$$

From the above, we immediately conclude that, for any $x \in OV_0, A \in SV_0$,

$$R_u(x,A) \Leftrightarrow (x\,\hat{\in} A) \notin \Gamma^* \text{ and } \neg(x\,\hat{\in} A) \notin \Gamma^*$$

Hence, we have:

Lemma 16.4
For any $x \in OV_0, A \in SV$, $R_t(x,A), R_f(x,A) R_u(x,A)$ are all mutually exclusive.

Proof. As Σ^* is not provable, then from Axiom 2 it follows that we cannot have both $(x\,\hat{\in} A) \in \Gamma^*$ and $\neg(x\,\hat{\in} A) \in \Gamma^*$. Hence $R_t(x,A), R_f(x,A)$ are mutually exclusive. The fact that $R_u(x,A)$ excludes both $R_t(x,A)$ and $R_f(x,A)$ follows trivially from its definition.

Define now

$$OV_1 = \bigcup\{\{x'_A, x''_A\} \mid x \in OV_0, A \in SV_0, R_u(x,A)\} \tag{16.7}$$

where

1. $OV_1 \cap OV_0 = \emptyset$
2. Any two elements in OV_1 represented by non-identical symbols are different (16.8)

As the universe of $\mathcal{A}$, we take

$$U = OV_0 \cup OV_1$$

Further, for any $x \in OV_0$,we define:

$$\begin{aligned} C_0(x) &= \{y \in OV_0 \mid (y \prec x) \in \Gamma^*\} \\ C_1(x) &= \bigcup\{\{x'_A, x''_A\} \mid A \in SV_0, R_u(x,A)\} \\ C(x) &= C_0(x) \cup C_1(x) \end{aligned} \tag{16.9}$$

Then, it can be easily seen that the family

$$\mathcal{C} = \{C(x) | x \in OV_0)\} \tag{16.10}$$

has the following properties:

Lemma 16.5. *For any $x, y \in OV_0$, the following holds:*

1. $x \in C_0(x)$, $C_0(x) \subseteq OV_0, C_1(x) \subseteq OV_1$;
2. *If $x, y \in OV_0$ and $x \neq y$, then $C(x) \cap C(y) = C_0(x) \cap C_0(y)$;*
3. $y \in C(x)$ *iff* $(y \prec x) \in \Gamma^*$;
4. $\mathcal{C}$ *is a covering of* U.

Proof.

1. Since by (16.5) x occurs in Σ^*, and Σ^* as a saturated sequent is closed under rule $(refl \prec)$, then $(x \prec x) \in \Gamma^*$, which yields $x \in C_0(x)$. The remaining two inclusions follow directly from (16.9), (16.5) and (16.7).
2. By (16.9), $C(x) = C_0(x) \cup C_1(x)$ and $C(y) = C_0(y) \cup C_1(y)$, where $C_0(x), C_0(y) \subseteq OV_0$, and $C_1(x), C_1(y) \subseteq OV_1$. As $OV_1 \cap OV_0 = \emptyset$ by (16.8), we have $C_0(x) \cap C_1(y) = C_1(x) \cap C_0(y) = \emptyset$. Moreover, from the definition of $C_1(x)$ in (16.9) and the fact that by (16.8) $\{x'_A, x''_A\} \cap \{y'_A, y''_B\} = \emptyset$ if $x \neq y$, we obtain $C_1(x) \cap C_1(y) = \emptyset$. Together with the foregoing, this yields $C(x) \cap C(y) = C_0(x) \cap C_0(y)$.
3. If $(y \prec x) \in \Gamma^*$, then $y \in C_0(x)$ by (16.9), whence $y \in C(x)$. Assume now $y \in C(x)$. If $y = x$, then $(x \prec x) \in \Gamma^*$, because Σ^* as a saturated sequent is closed under rule $(refl \prec)$. If $y \neq x$, then, as $y \in C(y)$ by Point 1, $y \in C(x)$ implies $y \in C(y) \cap C(x)$. As $C(y) \cap C(x) = C_0(y) \cap C_0(x)$ by Point 2, this yields $y \in C_0(x)$, whence $(y \prec x) \in \Gamma^*$.
4. To prove that $\mathcal{C}$ is a covering of U, we have to show that for every $u \in U$ there is $C \in \mathcal{C}$ such that $u \in C$. For $u \in OV_0$ this follows from Point 1. In turn, if $u \in OV_1$,

then $u \in \{x'_A, x''_A\}$ for some $x \in OV_0, A \in SV_0$. Thus from (16.9) we get $u \in C_1(x)$, whence $u \in C(x)$.

Accordingly, for our counter-model, we take $\mathcal{A} = (U, \mathcal{C})$ defined as above.

The valuation of object variables is defined as follows:

$$v(x) = x \tag{16.11}$$

for any $x \in OV_0$, and $v(x) = x_0$ for any $x \in U - OV_0$, where x_0 is some arbitrarily chosen, but fixed variable in OV_0.

To define the valuation of set variables, for any $A \in SV$ we take:

$$\begin{array}{ll} w_t(A) = \bigcup\{C(x) \mid R_t(x,A)\}, & w_f(A) = \bigcup\{C(y) \mid R_f(y,A)\}, \\ w'_u(A) = \{z'_A \mid R_u(z,A)\}, & w''_u(A) = \{z''_A \mid R_u(z,A)\}, \end{array}$$
$$w(A) = w_t(A) \cup w'_u(A) - (w_f(A) \cup w''_u(A)) \tag{16.12}$$

It remains to prove that $\mathcal{R}$ is indeed the desired counter-model, that is, that for the interpretation $I_{\mathcal{R}}$ of formulas generated by $\mathcal{R}$ according to Definition 16.7 we have

(I) $I_{\mathcal{R}} \models \Sigma$ for each $\Sigma \in S$;
(II) $I_{\mathcal{R}} \not\models (\Gamma \Rightarrow \Delta)$.

We start with (II). As $\Gamma \subseteq \Gamma^*, \Delta \subseteq \Delta^*$, then in order to prove (II) it suffices to prove that $I_{\mathcal{R}} \not\models (\Gamma^* \Rightarrow \Delta^*)$. Since the only designated value in the semantics of our logic is $\mathbf{t}$ and $I_{\mathcal{R}}(\varphi) \in \{\mathbf{f}, \mathbf{t}, \mathbf{u}\}$ for any formula φ, this means we have to show that

$$I_{\mathcal{R}}(\gamma) = \mathbf{t} \text{ for any } \gamma \in \Gamma^*, \qquad I_{\mathcal{R}}(\delta) \in \{\mathbf{f}, \mathbf{u}\} \text{ for any } \delta \in \Delta^* \tag{16.13}$$

For this purpose, we will prove, analogously as in [4], that, for any formula φ occurring in Σ^*,

$$(\mathbf{A}) \qquad I_{\mathcal{R}}(\varphi) = \begin{cases} \mathbf{t} \text{ if } \varphi \in \Gamma^* \\ \mathbf{f} \text{ if } \neg\varphi \in \Gamma^* \end{cases}$$

and

$$(\mathbf{B}) \qquad I_{\mathcal{R}}(\varphi) \in \begin{cases} \{\mathbf{f}, \mathbf{u}\} \text{ if } \varphi \in \Delta^* \\ \{\mathbf{t}, \mathbf{u}\} \text{ if } \neg\varphi \in \Gamma^* \end{cases}$$

We argue by induction on the complexity of φ, proving simultaneously **(A)** and **(B)**. For simplicity, from now on we drop the subscript $\mathcal{R}$ in $I_{\mathcal{R}}$.

$\varphi = (x \prec y)$, where $x, y \in OV_0$

(A) $\varphi \in \Gamma^*$
Then, $(x \prec y) \in \Gamma^*$. By Definition 16.7, to prove that $I(x \prec y) = \mathbf{t}$, we have to show that $v(x) \prec_{\mathcal{C}} v(y)$. Since by (16.11) $v(x) = x, v(y) = y$, in view of (16.4), this amounts to proving that, for any $C \in \mathcal{C}$, $y \in C$ implies $x \in C$. Suppose $C \in \mathcal{C}$ and $y \in C$. Then, by (16.10) $C = C(u)$ for some

$u \in OV_0$. Hence, by Point 3 of Lemma 16.5 $y \in C$ implies $(y \prec u) \in \Gamma^*$. However, as $(x \prec y) \in \Gamma^*$ too, and Γ^* is a saturated sequent, then by rule $(tr \prec)$ we have $(x \prec u) \in \Gamma^*$, whence $x \in C(u)$. As a consequence, $x \prec_C y$, whence $I(\varphi) = I(x \prec y) = \mathbf{t}$, and **(A)** holds for φ.

$\neg\varphi \in \Gamma^*$

In this case, $\neg(x \prec y) \in \Gamma^*$. As $S \not\vdash^{at}_{\mathrm{CRS}} \Sigma^*$, then by Axiom A2 $(x \prec y) \notin \Gamma^*$. Hence, by Point 3 of Lemma 16.5 $x \notin C(y)$. Since $y \in C(y)$, by (16.4), this implies $x \not\prec_C y$, whence $I(\varphi) = I(x \prec y) = \mathbf{f}$, and **(A)** holds for φ.

(B) $\varphi \in \Delta^*$

If $(x \prec y) \in \Delta^*$, then, as $S \not\vdash^{at}_{\mathrm{CRS}} \Sigma^*$, $(x \prec y) \notin \Gamma^*$ by Axiom A1 . Hence, from the immediately preceding proof of **(A)** in case of $\neg(x \prec y) \in \Gamma^*$we obtain $I(\varphi) = I(x \prec y) = \mathbf{f}$, and **(B)** holds for φ.

$\neg\varphi \in \Delta^*$

If $\neg(x \prec y) \in \Delta^*$, then, since Σ^* is a saturated sequent, from rule $(\Rightarrow \neg \prec)$ we obtain $(x \prec y) \in \Gamma^*$, whence $I(x \prec y) = \mathbf{t}$ by the inductive hypothesis for **(A)**. Consequently, $I(\varphi) = \mathbf{t}$ and **(B)** holds for φ.

$\varphi = (x \hat{\in} A)$, where $x \in OV_0, A \in SV_0$

(A) $\varphi \in \Gamma^*$

Then $(x \hat{\in} A) \in \Gamma^*$. By Point 2 of Definition 16.7, in order to show that $I(x \hat{\in} A) = \mathbf{t}$, we have to prove that $v(x) \in POS(w^*(A))$. Since $v(x) = x$ (for $x \in OV_0$) and $w^*(A) = w(A)$ because $A \in SV_0$, then by Equation (16.3) we need to prove that

$$(\forall C \in \mathcal{C})[x \in C \ \rightarrow \ C \subseteq w(A)]$$

By Equation (16.12), we have

$$w(A) = w_t(A) \cup w'_u(A) - (w_f(A) \cup w''_u(A)),$$

where

$$w_t(A) = \bigcup\{C(x) \mid R_t(x,A)\}, \quad w_f(A) = \bigcup\{C(y) \mid R_f(y,A)\}, \quad w'_u(A) = \{z'_A \mid R_u(z,A)\}, \quad w''_u(A) = \{z''_A \mid R_u(z,A)\} \tag{16.14}$$

Assume $C \in \mathcal{C}$ and $x \in C$. Then, by (16.10) and (16.9), there is $s \in OV_0$ such that $C = C(s)$, whence from by Point 3 of Lemma 16.5 we get $(x \prec s) \in \Gamma^*$. Considering that Σ^* is saturated and $(x \hat{\in} A) \in \Gamma^*$, from rule $(cl \ \hat{\in} \prec)$, we obtain $(s \hat{\in} A) \in \Gamma^*$. Thus, $R_t(s,A)$ by (16.6), whence (16.14) implies $C = C(s) \subseteq w_t(A)$. Accordingly, in order to prove that $C \subseteq w(A)$, we need to show that

(i) $z''_A \notin C(s)$ for any $z \in OV_0$ such that $R_u(z,A)$;

(ii) $C(s) \cap C(y) = \emptyset$ for any $y \in OV_0$ such that $R_f(y,A)$

We begin with (i). Assume $R_u(z,A)$. As $R_t(s,A)$ holds, and by (16.4) we cannot have both $R_t(s,A)$ and $R_u(s,A)$, then $z \neq s$. By (16.9), we have

$$C(s) = C_0(s) \cup C_1(s), \text{ where } C_1(s) = \{s'_B \mid R_u(s,B)\} \cup \{s''_B \mid R_u(s,B)\}$$

As $z''_A \in OV_1$ and $C_0(s) \subseteq OV_0$, then $z''_A \notin C_0(s)$. In turn, as $z \neq s$, then by (16.8) $z''_A \neq s'_B, s''_B$ for any B. Thus $z'' \notin C(s)$.
Consider now (ii), and assume $R_f(y,A)$. We argue by contradiction. Suppose $C(s) \cap C(y) \neq \emptyset$, and consider any $u \in C(s) \cap C(y)$. Then, by Point 3 of Lemma 16.5, we have $(u \prec s) \in \Gamma^*$ and $(u \prec y) \in \Gamma^*$. However, as $(s\ \hat{\in} A) \in \Gamma^*$ and $\neg(y\ \hat{\in} A) \in \Gamma^*$, this is a contradiction — for, in view of Axiom A3', any sequent containing all of the formulas $u \prec s, u \prec y, s\ \hat{\in} A, \neg(y\ \hat{\in} A)$ on the left hand side is provable in CRS^{at} by weakening, and Σ^* is not provable. Hence $C \subseteq w(S)$, $I(\varphi) = \mathbf{t}$ and **(A)** holds for φ.

$\neg\varphi \in \Gamma^*$
Then, $\neg(x\ \hat{\in} A) \in \Gamma^*$. By Point 2 of Definition 16.7, in order to prove that $I(\varphi) = I(x\ \hat{\in} A) = \mathbf{f}$, we have to show that $v(x) \in NEG(w(A))$. As $v(x) = x$, then by Equation (16.3), we need to prove that

$$(\forall C \in \mathcal{C})(x \in C \ \rightarrow\ C \subseteq U - w(A))$$

The proof is quite analogous to the preceding case. Assuming $x \in C$, we get $C = C(s)$ for some $s \in OV_0$, with $(x \prec s) \in \Gamma^*$ holding by Lemma 16.5, whence $\neg(s\ \hat{\in} A) \in \Gamma^*$ by rule $(cl\ \neg\ \hat{\in}\ \prec)$. By Equation (16.14), we have

$$U - w(A) = w_f(A) \cup w''_u(A) - (w_t(A) \cup w'_u(A)) \tag{16.15}$$

with $w_f(A), w_t(A), w'_u(A)$ and $w''_u(A)$ defined as in (16.14). Since $\neg(s\ \hat{\in} A) \in \Gamma^*$, then $R_f(s,A)$. As in the foregoing proof for $(x\ \hat{\in} A) \in \Gamma^*$ we have shown that $R_t(r,A)$ and $R_f(s,A)$ imply $C(r) \cap C(s) = \emptyset$, then to show that $C(s) \subseteq U - w(A)$ it suffices to prove that $z'_A \notin C(s)$ for any z such that $R_u(z,A)$. The proof is analogous to the proof that $z''_A \notin C(x)$ in case of $(x\ \hat{\in} A) \in \Gamma^*$. Hence, $C \subseteq U - w(A)$, $I(\varphi) = \mathbf{f}$ and **(A)** holds for φ.

(B) $\varphi \in \Delta^*$
Then $(x\ \hat{\in} A) \in \Delta^*$, whence $S \not\vdash^{at}_{\text{CRS}} \Sigma^*$ implies $(x\ \hat{\in} A) \notin \Gamma^*$ by Axiom A1. We have two possible cases:
$\neg(x\ \hat{\in} A) \in \Gamma^*$
Then $I(\varphi) = \mathbf{f}$ by what we have already proved in **(A)**, and **(A)** holds for φ.

$\neg(x\ \hat{\in} A) \notin \Gamma^*$
Then both $(x\ \hat{\in} A) \notin \Gamma^*$ and $\neg(x\ \hat{\in} A) \notin \Gamma^*$, whence $R_u(x,A)$. Accordingly, by what we have already proved for **(A)** in case of $(x\ \hat{\in} A) \in \Gamma^*$

and in case of $\neg(x\hat{\in}A)\in\Gamma^*$, $x'_A\notin C(r)$ for every r such that $R_t(r,A)$ and $x''_A\notin C(y)$ for every y such that $R_f(y,A)$. Since $x'_A\neq x''_A$, by (16.14) and (16.15) this yields $x'_A\in w(A)$ and $x''_A\in U-w(A)$. As $C(x)\supseteq\{x'_A,x''_A\}$, this implies $C(x)\cap w(A)\neq\emptyset$ and $C(x)\cap(U-w(A))\neq\emptyset$. Thus, by (16.3), we have $x\in Bnd(w^*(A))$, and so by Point 2 of Definition 16.7 $I(\varphi)=I(x\in A)=\mathbf{u}$. Hence, **(B)** holds for φ.

$\neg\varphi\in\Delta^*$

Then $\neg(x\hat{\in}A)\in\Delta^*$, whence $\neg(x\hat{\in}A)\notin\Gamma^*$. If $(x\hat{\in}A)\in\Gamma^*$, then $I(x\hat{\in}A)=\mathbf{t}$ by what we have already proved for **(A)**, and so $I(\varphi)=\mathbf{t}$. If $(x\hat{\in}A)\notin\Gamma^*$, then by what we have just proved above $I(x\hat{\in}A)=\mathbf{u}$, whence $I(\varphi)=\mathbf{u}$. In both cases, **(B)** holds for φ.

$\varphi=\neg\psi$

(A) If $\varphi\in\Gamma^*$, then $\neg\psi\in\Gamma^*$, whence by the inductive hypothesis for ψ, we get $v(\psi)=\mathbf{f}$, and from Point 3 of Definition 16.7, we obtain $v(\varphi)=v(\neg\psi)=\mathbf{t}$. Suppose now $\neg\varphi\in\Gamma^*$. Then $\neg\neg\psi\in\Gamma^*$, and, as $\Gamma^*\Rightarrow\Delta^*$ is saturated, by rule $(\neg\neg\Rightarrow)$ we have $\psi\in\Gamma^*$. Hence, by the inductive hypothesis $v(\psi)=\mathbf{t}$, which in turn yields $I(\varphi)=I(\neg\psi)=\mathbf{f}$ by Point 3 of Definition 16.7. Thus, **(A)** holds in this case too.

(B) If $\varphi\in\Delta^*$, then $\neg\psi\in\Delta^*$, whence by the inductive hypothesis for ψ, we get $I(\psi)\in\{\mathbf{t},\mathbf{u}\}$. By the definition of I, the latter implies $I(\varphi)=I(\neg\psi)\in\{\mathbf{f},\mathbf{u}\}$. If $\neg\varphi\in\Delta^*$, then $\neg\neg\psi\in\Delta^*$, and, as $\Gamma^*\Rightarrow\Delta^*$ is saturated, by rule $(\Rightarrow\neg\neg)$ we have $\psi\in\Delta^*$. Thus, by the inductive hypothesis $I(\psi)\in\{\mathbf{f},\mathbf{u}\}$, whence $I(\varphi)=I(\neg\psi)\in\{\mathbf{t},\mathbf{u}\}$ by the definition of I. Hence, **(B)** holds for φ.

$\varphi=\psi_1\vee\psi_2$

(A) If $\varphi\in\Gamma^*$, then $(\psi_1\vee\psi_2)\in\Gamma^*$. Since $\Gamma^*\Rightarrow\Delta^*$ is saturated, by rule $(\vee\Rightarrow)$, we have either $\psi_1\in\Gamma^*$ or $\psi_2\in\Gamma^*$. Hence, by the inductive hypothesis either $I(\psi_1)=\mathbf{t}$ or $I(\psi_2)=\mathbf{t}$. As in both cases, we get $I(\varphi)=I(\psi_1\vee\psi_2)=\mathbf{t}$, then **(A)** holds for φ.

If $\neg\varphi\in\Gamma^*$, then $\neg(\psi_1\vee\psi_2)\in\Gamma^*$. Since $\Gamma^*\Rightarrow\Delta^*$ is saturated, by rule $(\neg\vee\Rightarrow)$ we have $\neg\psi_1,\neg\psi_2\in\Gamma^*$, whence by the inductive hypothesis $I(\psi_1)=I(\psi_2)=\mathbf{f}$. By Point 3 of Definition 16.7, this yields, $I(\varphi)=I(\psi_1\vee\psi_2)=\mathbf{f}$, so **(A)** holds for φ.

(B) If $\varphi\in\Delta^*$, then $\psi_1\vee\psi_2\in\Delta^*$. Thus, as Δ^* is saturated, by rule $(\Rightarrow\vee)$ we have $\psi_i\in\Delta^*$ for $i=1,2$. Hence, by the inductive hypothesis $I(\psi_i)\in\{\mathbf{f},\mathbf{u}\}$ for $i=1,2$. As a result, by the definition of I, we have $I(\psi_1\vee\psi_2)\in\{\mathbf{f},\mathbf{u}\}$, so **(B)** is satisfied.

Next, if $\neg\varphi\in\Delta^*$, then $\neg(\psi_1\vee\psi_2)\in\Delta^*$. Since Δ^* is saturated, from rule $(\Rightarrow\neg\vee)$, we obtain $\neg\psi_i\in\Delta^*$ for some $i\in\{1,2\}$. Hence, by the inductive hypothesis $I(\psi_i)\in\{\mathbf{t},\mathbf{u}\}$ for some $i\in\{1,2\}$, and so $I(\psi_1\vee\psi_2)\in\{\mathbf{t},\mathbf{u}\}$ by the definition of I. Thus **(B)** holds for φ.

$\varphi = \psi_1 \wedge \psi_2$

The proof in this case is similar to that in the previous one and is left to the reader.

It remains to prove (I), that is, to show that $v \models \Sigma$ for each $\Sigma \in S$. So let $\Sigma \in S$. Then $\Sigma = \varphi_1, \ldots, \varphi_k \Rightarrow \psi_1, \ldots, \psi_l$ for some integers k, l and formulas $\varphi_i, \psi_j, i = 1, \ldots, k, j = 1, \ldots, l$. Clearly, we cannot have both $\{\varphi_1, \ldots, \varphi_k\} \subseteq \Gamma^*$ and $\{\psi_1, \ldots, \psi_l\} \subseteq \Delta^*$, for then $\Gamma^* \Rightarrow \Delta^*$ would be derivable from Σ, and hence, from S, by weakening. Since $F(S) \subseteq \Gamma^* \cup \Delta^*$, this implies that either $\varphi_i \in \Delta^*$ for some i or $\psi_j \in \Gamma^*$ for some j. Hence, by (A) and (B), which we have already proved, we have either $v \not\models \varphi_i$ for some i, or $v \models \psi_j$ for some j, which implies that $v \models \Sigma$.

16.9 Conclusions

In this chapter, we have presented a three-valued logic for covering-based rough sets, featuring additionally the subordination predicate which represents a fundamental relation between objects induced by the covering. The sequent calculus CRS developed in the chapter has been proved sound for the full language, and complete for the sublanguage $\mathcal{F}_{RS}^{at}$ of formulas containing only atomic set expressions.

The completeness proof for that sublanguage contains construction of the covering based on the subordination relation. However, unlike the indiscernibility relation underlying Pawlak's rough sets, the subordination relation does not uniquely determine the covering generating the approximation space — which might be one of the reasons for difficulties in proving completeness for the full language.

Another is certainly the need to handle the two cases of non-determinacy in the Nmatrix $\mathcal{M}_{RS}^3$. Because of them, in order to extend the completeness proof of the sublanguage given in the foregoing to the full language, we have to ensure in the counter-model construction, among others, the possibility that $I(x \,\hat{\in} (A \cup B)) = \mathbf{t}$ when $I(x \,\hat{\in} A) = I(x \,\hat{\in} B) = \mathbf{u}$. This amounts to enabling the option $v(x) \in Pos(A \cup B)$ when $v(x) \in Bnd(A) \cap Bnd(B)$. Since for the interpretation w^* of set expressions, we have $w^*(A \cup B) = w^*(A) \cup w^*(B)$, we cannot ensure that $C \subseteq w^*(A \cup B)$ for every $C \in \mathcal{C}$ such that $x \in C$ by adding to $w^*(A \cup B)$ anything which is not already included in either $w^*(A)$ or $w^*(B)$. Instead, we have to insert in $w^*(A)$ and $w^*(B)$ elements which will ensure that $v(x) \in Pos(A \cup B)$ while preserving the condition $v(x) \in Bnd(A) \cap Bnd(B)$.

This task is not so difficult in case of atomic A, B and isolated formulas of the type $x \,\hat{\in} (A \cup B)$. Indeed, assume in the proof of Theorem 1 that $\Gamma^* = \Gamma', x \,\hat{\in} (A \cup B)$, where A, B, x do not occur in Γ'. Then neither $x \,\hat{\in} A, x \,\hat{\in} B$ nor their negations are in Γ^*, so for the countermodel defined as in the discussed proof we have $I(x \,\hat{\in} A) = I(x \,\hat{\in} B) = \mathbf{u}$. Further, it is easy to check that for the covering $\mathcal{C}$ and the interpretation w of set variables defined there we have $C(x) = \{x, x'_A, x''_A, x'_B, x''_B\}$ and $w(A) = \{x'_A\}, w(B) = \{x'_B\}$. Now let us modify the definition of w given in the proof to w_m by taking $w_m(A) = \{x, x'_A, x''_B\}, w_m(B) = \{x'_B, x''_A\}$. Then we have $x \in Bnd(w_m(A))$, because $x'_A, x''_A \in C(x)$, but $x'_A \in w_m(A)$ and $x''_A \notin w_m(A)$, and analogously $x \in Bnd(w_m(B))$.

Hence, for the modified interpretation I_m generated by w_m, we have $I_m(x\ \hat{\in} A) = I_m(x\ \hat{\in} B) = \mathbf{u}$. However, as $w_m(A) \cup w_m(B) = \{x, x'_A, x''_B\} \cup \{x'_B, x''_A\} = C(x)$, then $x \in Pos(A \cup B)$, and so $I_m(x\ \hat{\in}(A \cup B)) = \mathbf{t}$.

However, the above task becomes quite daunting in case of nested set-theoretic operations and multiple formulas which must be handled simultaneously. Accordingly, after exploring the subject in depth, the author suspects that maybe no complete finite proof system exists for the language considered in this chapter.

Accordingly, the issue of a complete proof system for the full language will be the subject of further research, and in future work we will consider supplementing the language by other operators to ensure unique determination of the covering and completeness. Other directions for future work include considering logics for rough sets defined by partial orders, lattices, and so on.

References

1. Avron, A.: Classical Gentzen-type methods in propositional many-valued logics. In: Fitting, M., Orłowska, E. (eds.) Beyond Two: Theory and Applications of Multiple-Valued Logic. STUDFUZZ, vol. 114, pp. 117–155. Physica Verlag, Heidelberg (2003)
2. Avron, A.: Logical Non-Determinism as a Tool for Logical Modularity: An Introduction. In: Artemov, S., Barringer, H., d'Avila Garcez, A.S., Lamb, L.C., Woods, J. (eds.) We Will Show Them: Essays in Honor of Dov Gabbay, vol. 1, pp. 105–124. College Publications (2005)
3. Avron, A., Konikowska, B.: Multi-valued calculi for Logics based on Non-determinism. Logic Journal of the IGPL 13, 365–387 (2005)
4. Avron, A., Konikowska, B.: Rough sets and 3-valued Logics. Studia Logica 90(1), 69–92 (2008)
5. Avron, A., Lev, I.: Canonical Propositional Gentzen-Type Systems. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 529–544. Springer, Heidelberg (2001)
6. Avron, A., Lev, I.: Non-deterministic multiple-valued structures. Journal of Logic and Computation 15, 241–261 (2005)
7. Balbiani, P., Vakarelov, D.: A modal Logic for Indiscernibility and Complementarity in Information Systems. Fundamenta Informaticae 45, 173–194 (2001)
8. Banjeeri, M.: Rough sets and 3-valued Lukasiewicz logic. Fundamenta Informaticae 32, 213–220 (1997)
9. Demri, S., Konikowska, B.: Relative Similarity Logics are Decidable: Reduction to FO^2 with Equality. In: Dix, J., Fariñas del Cerro, L., Furbach, U. (eds.) JELIA 1998. LNCS (LNAI), vol. 1489, pp. 279–293. Springer, Heidelberg (1998)
10. Demri, S., Orłowska, E., Vakarelov, D.: Indiscernibility and complementarity relations in information systems. In: Gerbrandy, J., Marx, M., de Rijke, M., Venema, Y. (eds.) JFAK: Esays Dedicated to Johan van Benthem on the Ocasion of his 50-th Birthday. Amsterdam University Press (1999)
11. Deneva, A., Vakarelov, D.: Modal logics for local and global similarity relations. Fundamenta Informaticae 31(3-4), 295–304 (1997)
12. Duentsch, I., Konikowska, B.: A multimodal logic for reasoning about complementarity. Journal for Applied Non-Classical Logics 10(3-4), 273–302 (2000)

13. Iturrioz, L.: Rough Sets and Three-valued Structures. In: Orłowska, E. (ed.) Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa. STUDFUZZ, vol. 24, pp. 596–603. Physica-Verlag, Heidelberg (1999)
14. Kleene, S.C.: Introduction to Metamathematics, D. van Nostrad Co. (1952)
15. Konikowska, B.: A logic for reasoning about relative similarity. In: Orłowska, E., Rasiowa, H. (eds.) Special Issue of Studia Logica, Reasoning with Incomplete Information. Studia Logica, vol. 58, pp. 185–226 (1997)
16. Łukasiewicz, J.: On 3-valued Logic, 1920. In: McCall, S. (ed.) Polish Logic. Oxford University Press (1967)
17. Lin, T.Y., Cercone, N. (eds.): Rough sets and Data Mining. Analysis of Imprecise Data. Kluwer, Dordrecht (1997)
18. Øhrn, A., Komorowski, J., Skowron, A., Synak, P.: The design and implementation of a knowledge discovery toolkit based on rough sets — The ROSETTA system. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1. Methodology and Applications, pp. 376–399. Physica Verlag, Heidelberg (1998)
19. Øhrn, A., Komorowski, J., Skowron, A., Synak, P.: The ROSETTA software system. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 2. Applications, Case Studies and Software Systems, pp. 572–576. Physica Verlag, Heidelberg (1998)
20. Orłowska, E.: Reasoning with Incomplete Information: Rough Set Based Information Logics. In: Proceedings of SOFTEKS Workshop on Incompleteness and Uncertainty in Information Systems, pp. 16–33 (1993)
21. Pagliani, P.: Rough set theory and logic-algebraic structures. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis. STUDFUZZ, vol. 13, pp. 109–190. Physica-Verlag, Heidelberg (1998)
22. Pawlak, Z.: Rough sets. Intern. J. Comp. Inform. Sci. 11, 341–356 (1982)
23. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
24. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 29(3), 1–10 (1997)
25. Pawlak, Z.: Rough sets theory and its applications to data analysis. Cybernetics and Systems 29, 661–688 (1998)
26. Pomykała, J.A.: Approximation operations in approximation space. Bull. Pol. Acad. Sci. 35(9-10), 653–662 (1987)
27. Sen, J., Chakraborty, M.K.: A study of intenconnections between rough and 3-valued Lukasiewicz logics. Fundamenta Informaticae 51, 311–324 (2002)
28. Vakarelov, D.: Information systems, similarity relations and modal logics. In: Orlowska, E. (ed.) Incomplete Information: Rough Set Analysis. STUDFUZZ, pp. 492–550. Physica-Verlag, Heidelberg (1998)
29. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111(1-4), 239–259 (1998)
30. Yao, Y.Y.: On Generalizing Rough Set Theory. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 44–51. Springer, Heidelberg (2003)
31. Zakowski, W.: On a concept of rough sets. Demonstratio Mathematica XV, 1129–1133 (1982)
32. Zhang, Y.-L., Li, J.J., Wu, W.-Z.: On axiomatic characterizations of three types of covering-based approximation operators. Information Sciences 180, 174–187 (2010)
33. Zhu, W., Wang, F.-Y.: On three types of covering-based rough sets. IEEE Transactions on Knowledge and Data Engineering 19(8), 1131–1144 (2007)

Chapter 17
Music Information Retrieval in Music Repositories

Bożena Kostek

Abstract. This chapter reviews the key concepts associated with automated *Music Information Retrieval* (MIR). First, current research trends and system solutions in terms of music retrieval and music recommendation are discussed. Next, experiments performed on a constructed music database are presented. A proposal for music retrieval and annotation aided by gaze tracking is also discussed.

Keywords: Music information retrieval, decision systems, music classification, music annotation, gaze tracking.

17.1 Introduction

Audio-video information currently available online is measured in petabytes. Thus, to improve access conditions to vast resources containing this type of information, it is necessary to create meta-descriptions of multimedia files in an automatic mode.

Music Information Retrieval (*MIR*) is an interdisciplinary domain that focuses on automated extraction of information from audio signals and enables to search the indexed information [5, 12, 14, 17, 34, 35, 38, 39]. It must be stressed that in the last few years a great deal of progress has been made as regards the scientific studies on music information retrieval [42]. The results of global research concentrated on the practical use of technical implementations and systems applications. The ongoing research focuses for example on the improvement of the efficiency and effectiveness of music recognition (*e.g.,* in terms of performance). This is a crucial issue, especially in the context of online music services that attract millions of users. In its early stages, the primary focus of MIR was on applications that allowed to search for music information through QBH, *Query-by-humming/singing/whistling*. The more

Bożena Kostek
Multimedia Systems Department, Gdansk University of Technology
Narutowicza 11/12 Gdansk, Poland
e-mail: bozenka@sound.eti.pg.gda.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 463–489.
springerlink.com

advanced applications support *Query-by-example* searching. However, today the most important research (and applications) relates to the retrieval in which a *content-based* analysis is used. In particular, the retrieval of a musical style, genre, or music referring to the mood/emotions of a musical piece is called *Query-by-category: musical style, genre, mood/emotion* [1, 26].

The basis of music recommendation systems is constituted by metadata added to musical pieces so that a user can search a music database effectively. According to the ID3v1 (or higher) audio file format specification, these data are called *tags*. They are included in the files, and they refer to different aspects of music tracks (such as title, composer, length, etc.). However, there are also tags specifying the tempo or musical instruments of a musical piece, etc. [15].

In the MIR literature, there are three main approaches in terms of automatic music annotation [6, 8, 24, 30]. File annotation by means of automatic tag retrieval from databases, such as Gracenote or FreeDB, is the simplest method. However, such recommending systems lack the capability to suggest specific musical piece before they are annotated manually. Hence, in the second approach, individuals are employed to manually add tags to music files [30]. This method requires a large number of "experts" with musical background and is time-consuming. This method is also called *social tagging*, when a statistically significant number of people participate in the process. This method can, for example, take the form in which key words describing a musical piece are added by users. Nevertheless, it must be remembered that manual annotation can also be problematic in the context of various musical tastes and preferences, which can lead to a situation where the same track is assigned to different genres by individuals with diverse musical experience. That is why, it is often observed that users may never be able to objectively assign appropriate attributes to a given musical piece.

The third approach uses information based on a low-level description of music. Low-level descriptors are usually based on the MPEG 7 standard, Mel-frequency cepstral coefficients (MFCC's) or, finally, dedicated parameters suggested by researchers [11, 13, 15, 16, 17, 22, 32, 33]. Within this approach, feature descriptors are assigned to a given music excerpt in order to perform automatic annotation of a given piece. This is followed by a search for similarities within the genre to carry out automatic tagging. The selection and quality of the parameterization method depends on the algorithm used in automatic genre classification. Thus, the adequate selection of parameters, the algorithm optimization in terms of signal processing and data exploration techniques serve as key technologies that provide effective music tagging automatically.

Resulting from this description, the issues of retrieval and recommendation are interconnected. Manual annotation of musical pieces may be supported by the analysis carried out with regard to computer users' reactions to music they listen to. Currently, the technological potential supports gaze-tracking, in which objectivization of annotation process is possible by means of observing the level of the user's interest in the retrieved multimedia material.

A prototype device, named Cyber-eye, was constructed in the Department of Multimedia Systems (MSD) of the Gdansk University of Technology (GUT). The

device illuminates computer users' eyes by infrared (IR) light, and it acquires eye view for further processing. According to the definition given by the GazeGroup [40] "gaze tracking is the process of measuring the "Point of Regard" (PoR) or the"Line of Sight" (LoS) of the eye, and tracking it over time. This process can be divided into two subprocesses:

- eye tracking, that is, detecting and tracking eye features and movements,
- gaze estimation, that is, calculating the eye gaze from eye features" [40].

The Cyber-eye system is composed of hardware solutions and accompanying software that analyzes a user's activity during a given task [18, 19]. The architecture of the system is presented in Fig. 17.4, and its working principles in Section 17.5.2.

This article presents a research project in which the gaze-tracking system is used to develop a method of musical document ranking. The method is based on the analysis of how computer users perceive information visually.

Firstly, the article provides a review of selected solutions in the field of music recommendation along with the examples of such systems. Secondly, a music database designed for research purposes is presented. The database is used for experiments aimed at the analysis of music genre classification efficiency. In the experiments, a dual approach to musical piece annotation, that is, manual (a user selects a given element from a user interface with a mouse) and automatic (a file is automatically annotated by the gaze-tracking system) was also proposed. Hence, this study endeavors to answer if it is possible to perform automatic tagging in relation to a given musical parameter (*e.g.*, piece tempo), and whether it is linked with a subject's musical experience. Simultaneous collection of data that refer to options selected with a mouse was done to objectivize the study through finding a correlation between mouse-click selections and the results from the gaze-tracking system.

17.2 Review of Selected Solutions in Terms of Music Recommendation

System solutions and music recommendation services include social networking systems, Internet radio stations, and Internet music stores as well as research or experimental solutions which operate on small music databases. These last ones, chronologically earlier, allowed to test proposed solutions for music classification efficiency. In this case, one can talk about systems that are based on the aforementioned *content-based* analysis, that is, features of a given piece or music type assigned to a given composer or music genre.

Systems creating play lists compliant with a user's profile are used in applications and music recommendation services. The largest Internet radio stations based on what is called *collaborative filtering* were built around this approach. This term is understood as finding answers by questioning a community within social networking services. The activity is based on the identification of a group of individuals with similar tastes and musical preferences within a large user community and can

provide a list of suggested answers (presented hierarchically – from the most to the least accurate response). Lately, another term appeared, called *context-boosted collaborative filtering*, a hybrid approach that takes into account also content-based filtering. Algorithms, typically used in data exploration, such as the Pearson correlation coefficient, cosine measure or *k*-NN algorithm, that reflect the similarities among musical pieces contained in a database and a query submitted by a user, etc. are employed to search such dependency and similarity. The calculated numerical index of such measures enables to arrange responses (*e.g.*, play lists or answers to a query) in the order regarding their similarity.

17.2.1 Music Retrieval within a Content-Based Analysis

Systems or solutions based on music analysis use both low-level parameterization and meta-description. The selection of feature vectors (FV) that reflect the appropriate representation of musical data cannot be carried out effectively, unless tools supporting automatic music classification are also utilized. For this reason, the classification stage is as important as an adequately selected parameter vector.

Two types of descriptors are used for audio signals within the MPEG 7 standard [22]: *low-level descriptors* and *high-level descriptors*. The ISO/IEC 15938 standard provides only a definition of parameters while the algorithms calculating these quantities are not specified. The MPEG 7 standard contains 17 low-level parameters divided into six groups. In particular, the application of parameters from the *BasicSpectral* group as well as the *AudioPower* parameter is useful for classification. Since the parameters included in the MPEG 7 standard are very popular in audio data classification, their definitions will not be given here.

In the field of Music Information Retrieval, one can find several music collections (*e.g.* GZTAN, ISMIR2004, MIREX2005, Magnatune) that are often used to test the efficiency of solutions proposed by various researchers in terms of automatic music classification by means of *benchmarking* [42]. Some selected results of the efficiency of music classification obtained in recent years on the basis of these collections are presented in Table 17.1. It is worth mentioning that these collections are not consistent as they differ in the number of musical pieces, file format, bit resolution, number of genres, etc. Hence, such a comparison is not fully possible and justified. Characteristics of music collections, mentioned above, are as follows:

- GTZAN – au format, 22 kHz, monophonic signals, 30 s, 1,000 files, 10 music genres,
- ISMIR2004 – mp3 format, 44 kHz, stereophonic signals, full-length music tracks, 1,458 music files, 6 music genres,
- MIREX2005 – mp3 format, 44 kHz, mono, based on Magnatune corpus – full-length music tracks, mp3 and lossless, 15,000 music files,1100 albums, 10 genres
- Million Song Dataset based on Magnatune corpus service, 2,400 songs across 8 genres, mp3 and lossless.

Table 17.1. Examples of results in the field of the MIR music genre retrieval

Authors	Music collection	Efficiency [%]
Bergsta *et al.* [4]	GTZAN	82.50
Li *et al.* [20]	GTZAN	78.50
Lidy *et al.* [21]	GTZAN	76.80
Benetos, Kotropoulos [3]	GTZAN	75.00
Holzapfel, Stylianou [9]	GTZAN	74.00
Tzanetakis *et al.* [33]	GTZAN	61.00
Holzapfel, Stylianou [9]	ISMIR2004	83.50
Pampalk *et al.* [25]	ISMIR2004	82.30
Lidy *et al.* [21]	ISMIR2004	79.70
Bergstra *et al.* [4]	MIREX2005	82.23
Lidy *et al.* [21]	MIREX2005	75.27
Mandel, Ellis [23]	MIREX2005	78.81
Hughes *et al.* [10]	Milion Song Dataset	82.7

Numerous examples of music genre classification can be found in literature. Using the provided meta-data, Hughes et al. [10] automatically extracted a corpus of 2,400 songs across 8 genres from the custom multi-genre corpora from the Million Song Dataset [41], namely ambient, blues, classic-rock, classical, country, jazz, hip-hop, and techno. The average ROC value, which corresponds to the accuracy of a binary-classification experiment involving one genre vs. all the others, obtained in their classification experiments was 82.7%. Such studies were intensively carried out in the Department of Multimedia Systems, and the gained results are encouraging, as with a similar number of music database files they allowed to obtain results at the level of 80% (and higher) [15, 16]. Lately, a new database was created for the new contest (ISMIS 2011) on music genre recognition [17]. The dataset and feature vectors were prepared by the Multimedia Systems Department, GUT. The competition attracted very large interest among Data Mining and Music Information Retrieval community: 292 teams with 357 members had registered, 150 of them actively participated, submitting over 12.000 solutions in total, largely outperforming baseline methods. The best results obtained during the contest were by A. Schierz and M. Budka, and the classification rate was of 0.87507 [31].

17.2.2 Systems Using a Community Interaction

Social networking systems that use a community interaction are to a large extent based on users' profiles. It is usually achieved either by means of gathering indirect information, *for example*, by observing reviewed tracks in an e-store, analyzing musical pieces/albums display time, storing a user's shopping list, analyzing other social networking services or by obtaining direct information, that is, asking for the

musical piece assessment on the progressive scale, arranging the collection in the order of favorite tracks, or selecting a "better" musical piece from two presentations and creating a user's preference list. In order to ensure interaction between a user (mainly a broadly understood user's profile) and the system, technologies and applications have been created or adopted that automatically transfer data for analysis (*e.g.*, the *scrobbler* application described in Section 17.3.1).

An example of a new approach to music retrieval that makes use of community interaction is the field of *Music Emotion Recognition* or personalized music recommendation. The tests carried out within the field explain how the subjective nature of emotion perception must be taken into account when constructing automatic systems of music emotion recognition (MER) [27].

Examples of music recommendation systems based on social networking services will be presented in Section 17.3.

17.3 Examples of Music Recommendation Systems

In this Section, examples of music recommendation systems are discussed. First, systems that attract a large number of users are mentioned. Often, based on known services (*e.g.*, Last.fm [51] and YouTube), new ones are created that combine the functionality of two or more services (such as wykop.pl). For example, user's registered with Last.fm automatically obtain video clips from YouTube according to their musical taste.

17.3.1 Pandora

Undoubtedly, one of the best-known Internet radio systems recommending music is the personalized Pandora radio service. This is a content-based recommender, although the features are extracted by humans. It is said that the result is high-quality data, but poor scalability. It is based on the Music Genome project (USA, 2000) [54], in which around 400 parameters were initially set and constituted a kind of genotype of a given musical piece. Pandora Media service was established in 2000. Since 2005, the service has been operating as a free-of-charge Internet radio station. However, it is currently available only in the USA. The music catalogue is estimated to contain 74000 pieces. The service was initially using seven music genres categorized into five groups (Pop/Rock, Hip-Hop/Electronica, Jazz, World Music, and Classical). Currently, there are 100 radio stations defined by listeners where the assigned music genre does not always conform with the ID3v2 format.

Each musical piece in the database is represented by a vector of attributes. The system uses from 150 to 500 attributes depending on the type of music genre. The attributes relate to music structure elements (*e.g.*, dynamics, tempo, and rhythm),

instruments, or lyrics. Musical pieces are assessed by experts (musicians and musicologists) and attributes which create an adequate genotype for a given track and a music profile are also assigned in this way. The analysis of a single musical piece takes about half an hour.

Pandora uses a feedback relying on a user's assessment of a musical piece for the purposes of efficient recommendation. The service uses the simplest form of subjective assessment: users specify whether they like a track or not. Listening to a musical piece suggested by the service may be skipped with no limitation. Opting not to make choice is interpreted as not being in the mood to listen to a given musical piece. A list of recommended tracks is defined on the basis of similarity vectors. The similarity is calculated as the proximity of vectors in a given metric space. The choice of a track or an artist *on demand* is not possible in this system [54].

17.3.2 Last.fm

Last.fm is currently the biggest musical social networking service [51]. It was created in 2002 by F. Miller, M. Stiksel, M. Breidenbruecker and T. Willomitzer. In 2005, it merged with the *Audioscrobbler* project by R. Jones from the University of Southampton. Presently, the service is managed by the CBS Corp. media company. Initially the service was free of charge. Currently, however, a monthly subscription is required. It is estimated that the community around Last.fm includes 40 million users, and the music catalogue contains about seven million pieces and is still growing. Moreover, the service has over 43 million registered *scrobblers*. Scrobbler is an application installed on a user's computer to download information from the client application, in this case from a multimedia player. This message is sent by Scrobbler to the service. Primarily, the service fulfils the role of an Internet radio station. It also recommends music and provides current rankings of artists, tracks, and the most popular tags. The service also provides information about music events. Geolocation takes place on the basis of the user's IP address.

Last.fm automatically generates a profile page for every user who signed up for a music recommender system called "*Audioscrobbler*". The system builds a detailed profile of each user's musical taste by recording details of the songs the user listens to, either from Internet radio stations or the user's computer or portable music devices. This information is transferred to the database (*scrobbled*) either via the music player itself or via a plugin installed in the user's music player. The profile data are then displayed on the user's profile page along with the list of the tracks recently listened to as well as a list of the artists most often listened to and the user's favorite tracks. On the basis of these data, the system retrieves and creates a list of other users – individuals with similar tastes. User pages include what is known as taste-meters that enable to compare the compatibility of different users' musical tastes. Additionally, a user can get information about which tracks have recently been listened to by his/her friends, etc.

Aforementioned, there are two basic ways of obtaining information on the preferences of particular users in the service:

- Using the Internet radio station in the service. A user may use the list of artists and tracks recommended by the service. A user can also use several radio channels. Every channel includes a separate play list (27 stations that are divided into the music of the sixties, seventies, eighties, and nineties).
- It is possible to build a user's profile on the basis of the tracks that were listened to as well as their rating. In this case, scrobbling allows obtaining information about a user by collecting information on the music database stored in a user's computer or by analyzing musical pieces that the user listens to. Scrobbling of a track occurs when the track is listened to. The title of the musical piece is sent to Last.fm where it is added to the user's music profile. These data enable to create personalized radio stations. Collaborative filtering is used in the service. Music suggestions are retrieved thanks to service users who have similar tastes (known as neighbors).

The users can tag artists, albums, and tracks, and thus retrieve music according to assigned tags. Besides the most typical choices (*e.g.*, a music genre, artist's or composer's name), the classification of this kind includes the choice of a track according to assigned mood, the artist's sex, or other characteristics. Initially, users were able to create custom radio stations and playlists from any of the audio tracks in the Last.fm music library and could listen to some individual tracks on demand. However, the ability to listen to custom radio stations ("personal tag radio") was withdrawn in 2010. An opportunity to create a radio station based on a given tag was possible, if the number of musical pieces that were annotated by this tag was sufficiently large. The service also allows to retrieve lyrics. However, as pointed out before, in 2010 some of the above-mentioned functions were limited [51].

17.3.3 Examples of Other Systems

Mufin

Music FINder – the system uses the mufin player (only on PC) [52]. The service is a search engine with recommendation and enables the users to buy music or listen to half-minute excerpts. The smartphone application is available with the Android system. The system is a classic example of content-based music recommenders, using an algorithmic approach. Recommendation occurs on the basis of musical piece parameters (tempo, instruments, harmony, etc.). The service presents musical pieces collected on the server graphically in the form of a three-dimensional map. This enables one to observe the relationships among particular tracks of the database. A user is able to define

- on the horizontal plane – the mood,
- on the vertical plane – sound choice (from synthetic to acoustic),

- when moving backwards or forwards, a user is able to select the adequate tempo of a music (from calm to aggressive).

Genius

The Genius function introduced in iTunes v8. allows recommending music among tracks available in the user's library [45]. Retrieval and recommendation use typical collaborative filtering, based, however, on the similarity of artists' (not songs or albums). The resulting recommendations take the form of a playlist of musical pieces that are similar or mixed consistently. The Genius Sidebar recommendation application uses Gracenote MusicID to collect high-level data.

Musicovery

Musicovery is an interactive Internet radio station created by Castaignet and Vavrille in 2006 in France [53]. The system creates playlists containing recommended tracks according to the user's mood. The free-of-charge use of the service allows to listen to low-quality files (32 kb/s). Experts who rate particular musical pieces are employed by the service. Each musical piece in the database is described by 40 attributes. The attributes subjectively describe only the *mood* of a song/music. Playlists of the user's anticipated preferences are built on the basis of the obtained classification. It is possible to select musical pieces from a specified decade. The service displays recommendation in the form of a map that shows the relationships among musical pieces.

Other systems that recommend music include

- **Lala –** based on "following" the P2P service users and their music choices, that is, the service relies on users following each other on the service and recommending new music to each other [50],
- **Amazon –** Internet store, the users may listen to or download music in their Cloud Drive accounts, also it lets users store music on remote servers and access songs from the internet or Android phones [46],
- **eMusic –** the second largest Internet music store after iTunes, based on the technology of personalization and recommendation services, takes into account a hybrid approach (algorithms and human input from experts) [47],
- **iLike –** part of MySpace; the service lets users share music preferences, receive personalized concert and music recommendations and includes a "sidebar" for iTunes that creates automatic playlists [48],
- **Slacker –** personalized Internet radio station (operating only in the USA) [55],
- **Jamendo –** the last of the presented systems is currently often compared with Last.fm. The service does not have the scrobbling function, so music can only be listened from the service site or the whole album can be downloaded [49].

17.4 Music Genre Recognition Experiment

17.4.1 Music Database

The music database created in the Department of Multimedia Systems contains musical pieces of 60 composers/artists. From the albums within the same style, 15–20 musical pieces of each musician/composer have been chosen. They were of CD-quality, standard 16-bit resolution/44.1 kHz sampling rate, PCM stereo format. Music genre categories in the created database were limited to basic styles, such as classical music, jazz, blues, rock, metal, and pop. The number of musical pieces assigned to a given genre is presented in Table 17.2. For each piece, 20 segments of 25-second-long-excerpts were extracted and parameterized. This means that 20 feature vectors containing 171 parameters were obtained for every musical piece. The parameterization involved extraction of the following parameters: 127 descriptors of the MPEG-7 standard, Mel-Frequency Cepstral Coefficients (MFCC – 20 descriptors) as well as time-related "dedicated" parameters (24 descriptors). For the MPEG-7 descriptors, the Hamming window of length 8192(with 50% overlapping) was used. For example, for a 5-minute audio excerpt, the number of frames was 155 (each audio segment 15-seconds long at a sample rate of 44100 Hz with overlapping results in 661500 samples; thus, this returns 155 frames). In the case of MPEG-7 descriptors the average and variance values were calculated within each audio segment for all frames containing 8192 samples over all frequency bands. For the MFCC parameters, the Hamming window of length 512 (with 50% overlapping) was utilized. The parameter vector includes 171 descriptors, and the total number of parameter vectors in the whole database amounts to 21,680. Additionally, it must be added that the parameters were normalized to be in the range of $(-1, +1)$. Since MPEG-7 features and Mel-Frequency Cepstral-Coefficients are widely presented in a rich literature related to this subject, there they will only be listed. These parameters are listed below:

- parameter 1: Temporal Centroid (defined as the time averaged over the energy envelope),
- parameter 2: Spectral Centroid (computed as the power weighted average of the frequency of the bins in the power spectrum) average value calculated for all frames (see comment above),
- parameter 3: Spectral Centroid variance calculated for all frames,
- parameters 4–37: Audio Spectrum Envelope (ASE)- describes the spectrum of the audio according to a logarithmic frequency scale; average values in 34 frequency bands (calculated at one-fourth-octave intervals over the range of 62.5 Hz to 16 kHz),
- parameter 38: ASE average value calculated for 34 frequency bands,
- parameters 39–72: ASE variance values in 34 frequency bands (the same frequency bands as cited above),
- parameter 73: averaged ASE variance parameter,

- parameters 74, 75: Audio Spectrum Centroid (defined as the power weighted log-frequency centroid) – average and variance values,
- parameters 76, 77: Audio Spectrum Spread (describes the second moment of the log-frequency power spectrum) – average and variance values,
- parameters 78–101: Spectral Flatness Measure (SFM) average values for 24 frequency bands, calculated at one-fourth-octave intervals over the range of 250 Hz to 16 kHz; (SFM describes the flatness properties of the spectrum of an audio signal within a given number of frequency bands),
- parameter 102: SFM average value (averaged for 24 frequency bands),
- parameters 103–126: Spectral Flatness Measure (SFM) variance values for 24 frequency bands,
- parameter 127: averaged SFM variance parameters (averaged for 24 frequency bands),
- parameters 128–147: 20 first MFCC (mean values),
- parameters 148–171: dedicated parameters in the time domain based on the analysis of the envelope distribution in relation to the rms (root mean square) value.

The dedicated parameters are related to the time domain. They are based on the analysis of the distribution of sound sample values related to the multiples of the root mean square values of the signal (rms). For this purpose, three reference levels were defined: r_1, r_2, and r_3 – equal to namely $1 \cdot rms$, $2 \cdot rms$, and $3 \cdot rms$ values of the samples in the analyzed signal frame. The first three parameters are linked with the number of samples that are exceeding the levels: r_1, r_2, and r_3.

$$p_n = \frac{count(samples\_exceeding\_r_n)}{length(x(k))} \tag{17.1}$$

where $n = 1,2,3$, and $x(k)$ is the signal frame analyzed.

The initial analysis of the values of parameters p_n showed to be difficult, because the rms level in the analyzed excerpts sometimes significantly varies within the frame. In order to deal with this problem, another approach was introduced. Each 5-second frame was divided into 10 smaller segments of 0.5 second. In each of these segments, the $p_n$parameters (Eq. 17.2) were calculated. As a result, a sequence of P_n was obtained:

$$P_n = \{p_n^1, p_n^2, p_n^3, \ldots, p_n^{10}\} \tag{17.2}$$

where p_n^k, $k = 1 \ldots 10$, and $n = 1,2,3$ as defined in Eq. 17.1.

In this way, 6 new features were defined on the basis of the P_n sequences. New features were defined as the mean (q_n) and variance (v_n) of P_n, $n = 1,2,3$. Index n is related to the different reference values of r_1, r_2, and r_3.

$$q_n = \frac{\sum_{k=1}^{10} p_n^k}{10} \tag{17.3}$$

$$v_n = \mathrm{var}(P_n) \tag{17.4}$$

In order to supplement the feature description, additional three parameters were defined. They are calculated as the "peak to rms" ratio, but in three different ways described below:

- parameter k_1 calculated for a 5-second frame,
- parameter k_2 calculated as the mean value of the ratio calculated in 10 sub-frames, each of 0.5-second duration (no overlap was used),
- parameter k_3 calculated as the variance value of the ratio calculated in 10 sub-frames (no overlap was used).

The last group of the dedicated parameters is related to the observation of the threshold crossing rate value. This solution may be compared with a more general idea based on a classical zero crossing rate parametrization (ZCR). The ZCR parametrization is widely used in many fields of automatic sound recognition. The extension of this approach is a definition of a threshold crossing rate value (TCR) calculated analogically as ZCR, but by counting the number of signal crossings in relation not only to zero, but also to the r_1, r_2 and r_3 values. These values (similarly as in the case of the other previously presented parameters) are defined in 3 different ways: for an entire 5-second frame and as the mean and variance values of the TCR calculated for 10 sub-frames. This gives 12 additional parameters to the feature set.

The entire set of dedicated parameters consists of 24 parameters that supplement 147 parameters calculated based on MPEG-7 and mel-cepstral parameterization.

The reduction in the feature vector redundancy is an important part of the analysis. Therefore, the obtained parameter vector was tested in the context of separability. The basic correlation analysis is based on calculating the covariance matrix followed by that of the correlation matrix and the interpretation of particular coefficients within the t-distribution statistics. Based on that it can be specified which parameters can be viewed as redundant. This may be done by means of the correlation coefficient R_{xy} calculated for values $x_1, x_2 \ldots x_n$ of parameter x and for values $y_1, y_2, \ldots y_n$ of parameter y according to the following formula:

$$R_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \tag{17.5}$$

where n is the number of parameter vectors. The t-distribution characterized by $n-2$ degrees of freedom is calculated according to the following formula:

$$\frac{R_{xy}}{\sqrt{1 - R_{xy}^2}}\sqrt{n-2} \tag{17.6}$$

In Figs. 17.1 and 17.2 examples of analyses are presented: the first one shows the report agreement in terms of the parameter correlation of the artist's songs with the parameters of the whole music category, while the second one shows the lack of such a report agreement.

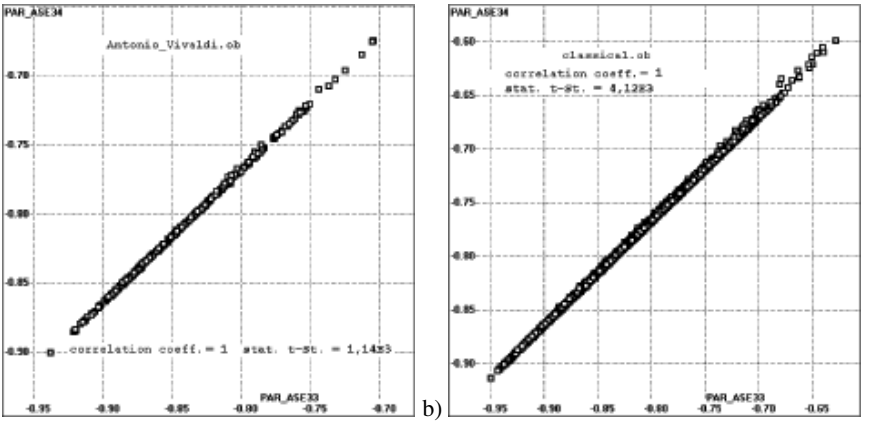


Fig. 17.1. Examples of graphs displaying the distribution of strongly correlated parameters with regard to the class of a selected artist (a) and the whole class of a music category (b)

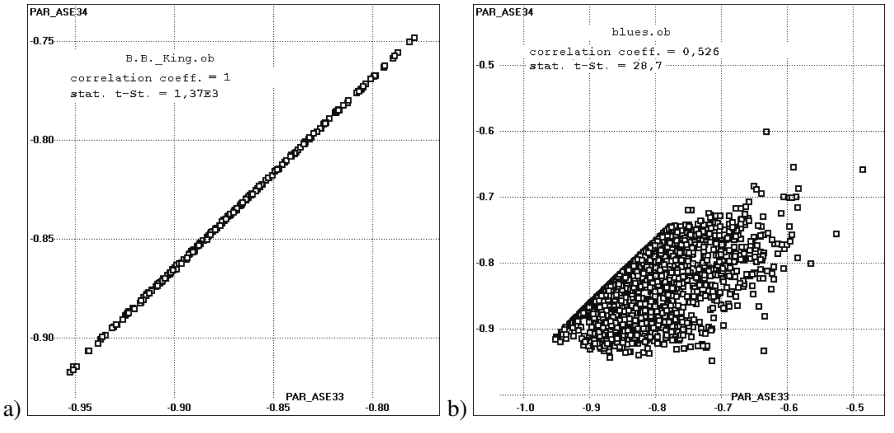


Fig. 17.2. Examples of graphs displaying the distribution of strongly correlated parameters with regard to the class of a selected artist (a) and the whole class of a music category (b)

Table 17.2. Presentation of the number of genres and musical pieces in the constructed music database

Music genre	Number of tracks
classical	320
jazz	362
blues	216
rock	124
metal	104
pop	184

17.4.2 Music Genre Classification

The research experiment included a study of the created parameter vector efficiency for automatic music classification. In order to achieve it, the data were divided into training and testing sets: 50/50%. The following algorithms were used for classification: *Support Vector Machine* – SVM, J48 decision trees and the Random Forest (RF) from the WEKA system [44], and the rough set-based (RS) analysis from the RSES system [43]. The C-SVC SVM algorithm and a Gaussian function with the Radial Basis kernel function were employed. The cost coefficient was set at 62.5 and the gamma parameter at 0.5. The J48 classifier parameters were default, so the Confidence Factor was 0.25, and the minimum number of instances per leaf was 2. The Random Forest model was created with the unlimited depth of trees and the number of trees, was equal to 20. For the J48, the confidence factor used for pruning was set to 0.25, and the minimum number of instances per leaf was equal to 2.

Rough sets introduced by Pawlak [28] are often employed when discovering significant and eliminating redundant data is a crucial aim. A large number of references focusing on rough sets-based applications [28, 29], also those related to MIR [13, 15] exist. Within the context of this chapter, the rough set method was used for the purpose of classification, even though rough sets - as mentioned above - are typically used to discover data dependencies and to identify data redundancy. In the experiments, the rough set decision system RSES was employed [2, 43]. Since this system is used by many researchers, the details concerning its algorithmic implementation and performance will not be provided here. FVs were divided into training and testing sets. The parameters were quantized according to the RSES system principles. The local and global discretization were used to obtain reducts calculated from genetic and exhaustive algorithms [2, 36, 37]. The exhaustive (deterministic) algorithm constructs all reducts, whereas the genetic (heuristic) algorithm calculates some of reducts [36, 37].

Since the classifier based on the Support Vector Machine is by definition a binary classifier, in the case of music genre classification, it required extension to classify many classes. In this case, the *one-versus-all* method was employed. This method involved creating many binary classifiers (the number of classifiers equals the number of classes), each one of them differentiating a class from the set of classes.

The classification was performed in the RSES [43] and WEKA environments [44]. Table 17.3 and Fig. 17.3 show results of music database classification.

Table 17.3. Classification efficiency

Classifier	Efficiency [%]
SVM	90.87
J48	77.40
Random Forest	84.72
Rough Sests	78.8

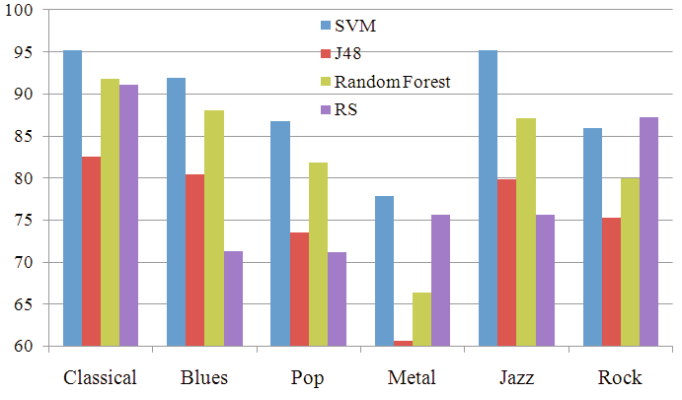


Fig. 17.3. Classification efficiency in terms of particular music genres

The data classified were strongly non-linear. It is worth noting that the SVM classifier looks for a hyperplane that divides classes while maintaining a maximum possible margin of distance from vectors at class borders. In the case of SVM, linear discrimination is realized in a plane that is different from the original one, which refers to the fact that the division border in the primary representation plane may be strongly non-linear. Considering this fact, it seems that the SVM classifier provides the greatest efficiency of recognition.

In addition, it is interesting to observe more detailed classification results produced by rough sets (Table 17.4). It seems that in both cases (SVM and RS) the set of parameters contained in the FV is very efficient in separating classical music from other genres. From the confusion matrix (Table 17.4), we can see that the optimization of feature vectors could be done for jazz genre, since this is often erroneously recognized as classical music.

Table 17.4. Classification efficiency

	Classical	Jazz	Rock	Blues	Metal	Pop	No. of objects	Accuracy	Coverage
Classical	2169	159	5	39	0	8	2380	**0.911**	1
Jazz	572	2110	19	149	15	95	2960	**0.713**	1
Rock	18	106	1097	49	129	141	1540	**0.712**	1
Blues	5	21	27	726	33	148	960	**0.756**	1
Metal	3	13	63	40	545	56	720	**0.757**	1
Pop	6	34	73	15	20	1012	1160	**0.872**	1
True positive rate	0.78	0.78	0.78	0.78	0.78	0.78			

17.5 Proposal for Objectivization of Annotation Process

The following factors were taken into consideration in the proposal for the objectivization of the annotation process:

- The way content is perceived which is evaluated through the degree of a user's concentration on a particular screen area, based on the heat-map obtained in the gaze-tracking study,
- Correspondence between the gaze-tracking point (automatic) and mouse (manual) selections.

It was assumed that the file annotation is done automatically by the gaze-tracking system. However, a user may also use a mouse in order to select an element in a specially prepared form (manual selection).

17.5.1 Description of Experiment Setup

Before subjects started the experiment, they were asked to fill in a form on musical preferences. The form included the subject's age, favorite music genres as well as the question whether they were musicians or if they listened to music regularly. This information allowed for dividing the subjects into four categories (musicians regularly listening to music, non-musicians regularly listening to music, etc.). Such a separation enabled to assess whether musical experience affects the subjects' capability to tag musical pieces. Thirty music files were used in the experiments, five from each of the six music genres included in the prepared database (jazz, blues, pop, rock, metal, classical music). The order of presenting a particular track was random. All the files constituted over half-a-minute-long excerpts of musical pieces in mp3 quality with a 128 kbps bit rate. Music was presented sequentially, but a user could listen to an excerpt limitlessly while tagging an individual musical piece. There was no time limit to run the experiment either. Furthermore, subjects were not informed about the role of the Cyber-eye system in the experiment, so that more objective results could have been obtained.

Apart from specifying a music genre subjects were also asked to classify other attributes of a given musical piece, that is, the type of the piece (*e.g.*, instrumental, choral, orchestral), its dynamics, tempo (*e.g.*, very slow, slow, ..., fast, ...), type of the recording (concert, studio, demo), and the mood of music (*e.g.*, neutral, sad, calm). The user could also indicate groups of instruments used in the recording. The obtained data include diverse information about files that were subject to such tagging.

17.5.2 Platform for Running the Experiment

A special questionnaire was designed in the form of a website within PHP + MySQL + jQuery technologies in order to run the experiment. Additionally, a modified version of WWW Cyber-eye application became the environment for running the experiment. The Cyber-eye system is based on infrared light (IR) and enables to measure gaze position.

The gaze-tracking system (Cyber-eye) system has already been described in other papers [18, 19], thus only its main features will be recalled here. The Cyber-eye system uses infrared (IR) illumination and allows to track the user's gaze position on a computer screen. It was constructed in compliance with the infrared illumination usage standards and is safe. The IR illumination is emitted by sections of IR LEDs placed in four corners of the display. This enables to detect a fixation point more accurately, because the estimation of the pupil shape and the determination of the pupil center directly associated with the gaze point, is also more precise. Secondly, the IR sources placed in the corners of the screen produce unique corneal reflections called glints on the eye cornea. They form a shape of quadrangle because they are created by four sections of LEDs located in four display corners (see Fig. 17.4).

Also, the localization of glints is characteristic. They are localized in the iris and the pupil. The pupil is always very bright because of the IR LED illuminator beaming light along the camera optical axis. It is worth to mention that section diodes along the camera axis cause the fifth glint appearance which is very useful in gaze estimation. Overall, contrast between the brightness of the iris and the pupil is relatively large, therefore finding an area with glints is possible. The image processing algorithm analyzes each frame, locating glints. The coordinates of the characteristic points (the pupil center and four glints) are sufficient to determine the point at which the user is looking. The gaze-tracking system is sufficiently robust in case the head of the subject is moving. Thus, a person is not obliged to keep his/her head in a fixed position. However, the system requires a very short calibration phase which is performed by a user using a specific application. This means that a user is asked to look at a series of points displayed sequentially on the computer screen and to gaze at these points for a few seconds. Such a time is sufficient to calibrate the system, that is, to determine the spatial variation between individual gaze samples. The whole calibration process takes approximately 1-2 minutes. To summarize, the information gathered by Cyber-eye provides a direct mapping between a "target point"-the area

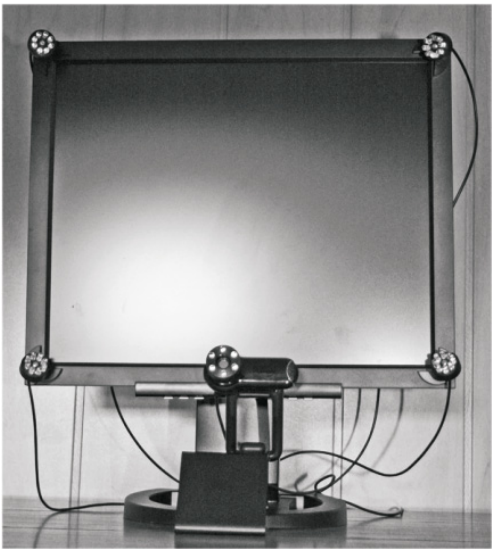


Fig. 17.4. Hardware configuration of the Cyber-Eye developed in the MSD, GUT

at which a user should be looking and the actual gaze point location. As a result of working with Cyber-eye heat maps, gaze plots, and fixation time, that is, time of gazing at a target site are registered [18]. A heat map is a two-dimensional representation of data in which values are represented by colors. Generated colors-from blue, the most infrequent to red, the most frequent - denote frequency of looking at the objects in the image. To simplify, the larger and red (more focused) the area in the heat map is, the longer the user fixated his/her gaze on a target. A gaze plot represents a visualization of the path a user's eyes followed from one point to another [7]. A line is drawn across the stimulus to represent the path, and circles of varying sizes signify "fixations" – areas where the user looked for a significant amount of time. Larger circles represent longer fixation time. An example of such a graph with a heatmap and gaze plot placed on the user's interface is presented in Fig. 17.5.

The design of the form was enforced by the display resolution that is, 1280×1024 and characteristics of the Cyber-eye system. The division of the form into three subpages was enforced by a larger than usual element size, as well as the necessity to create a page without the necessity to scroll down its contents. The three subpages are as follows: genre, music type (page No. 1), dynamics, tempo, recording type (page No. 2), mood, instruments used (page No. 3). To facilitate navigation, a constant element in the shape of a time line of a given musical piece as well as navigation elements through the mentioned pages were displayed in the upper part of the form on each page.

All choices made by subjects and music preferences provided in the questionnaire were saved in a MySQL database on the server. WWW Cyber-eye provided additional information in the form of xml files that included a list of selected HTML

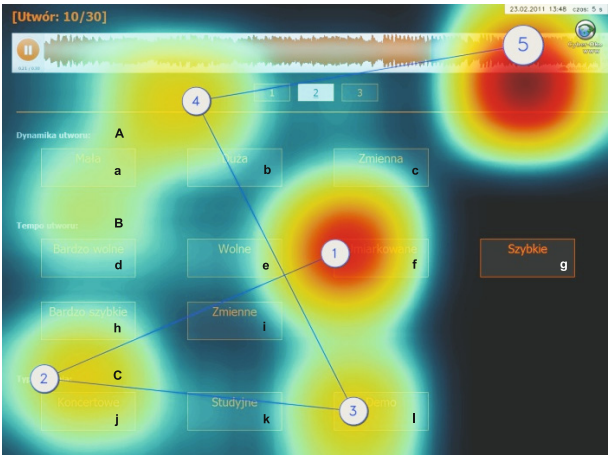


Fig. 17.5. The heatmap generated by WWW Cyber-eye on the designed interface - page No. 2 (*in Polish*); denotations are as follows: A - dynamic range, B - tempo, C- type of recording, a - low, b - high, c - changeable, d - very slow, e - slow, f - moderate, g - fast, h - very fast, i - changeable, j - concert-live, k - studio, l - demo

elements as well as time periods spent on looking at individual elements (with resolution of 0.1 s). Data for each musical piece were generated for every subpage. Musical pieces used in the experiments had selected tags, *for example* the artist's name, song or album titles along with the assigned genre. This information was obtained on the basis of Gracenote collection. However, data were not presented to users to avoid suggestion. The platform used to store and play music files was Soundcloud, integrated with the engineered application.

17.5.3 Annotation-Related Experiment

The experiment was performed in the laboratory of the Department of Multimedia Systems. Six computers equipped with Cyber-eye and headphones used to listen to music excerpts were employed. A group of 20 individuals (between 22 and 30 years old) was tested. After the initial Cyber-eye calibration, all subjects filled in a musical preference questionnaire and then tagged individual musical pieces. The initial questionnaire helped to divide subjects into the categories indicated in Table 17.5.

Since the experiment tested numerous aspects of tagging, only some chosen results are presented below. When analyzing the results, several aspects were taken into consideration, such as for example, questions about the type of music or tagging the tempo of a musical piece. The conclusions are based on the comparison between selections done by clicking and those achieved from the analysis of gaze tracking.

Table 17.5. Group size

No. of group	Groups of subjects and their numbers		
	Musician	Listener	Number of subjects
1	Yes	Yes	8
2	Yes	No	1
3	No	Yes	9
4	No	No	2

Since, on the basis of the initial questionnaire, every participant was classified into one of the four groups, only some representatives of the groups were selected for analysis. Subjects were required to select their favorite genres in the initial questionnaire. In Table 17.6, preferences of music genres among subjects are presented. It shows the number of individuals that selected a given genre. All the subjects that claim to listen to music regularly (3 of 20 claim not to listen to music) chose rock as their favorite genre.

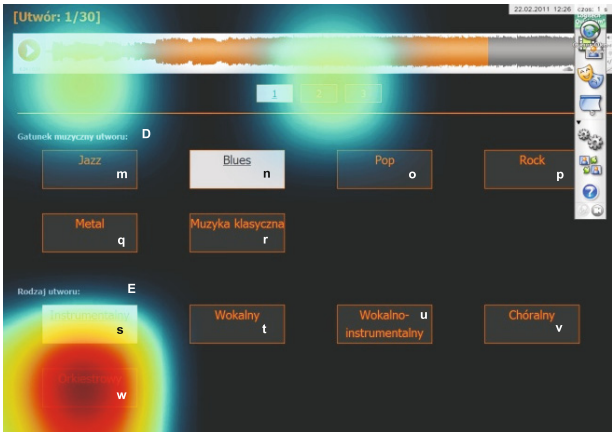
Table 17.6. Preferences of music genres among subjects

Music genre	Number of subjects
Jazz	4
Blues	2
Pop	7
Rock	17
Metal	6
Classical	5

An example of the heat map generated for two pages No. 1 and 2 constructed is shown in Figure 17.6. Figures 17.6a and 17.6b illustrate that questions about the type of music or recording were relatively easy to answer for subjects even in the case when a music piece was not their favorite genre (excerpt from Fig. 17.6a refers to blues, and from Fig. 17.6b to metal). It should also be noted that the results shown in Fig. 17.6 refer to the first and the last excerpts presented to the subject thus in this case there was not a problem with familiarization with the task to be performed. Based on the analysis performed for these two sub-pages, it may be reported that for the type of the excerpt or recording there is a clear correlation between the one given by the tag, indicated by the mouse click and the one by the gaze fixation. In this case, there were only few misclassifications among subjects.

However, it occurs that the most difficult task was to tag tempo. The first problem encountered might be probably related to the lack of familiarization with the sub-page constructed. In Fig. 17.7, one can easily observe that while listening to song No. 1 the subject's gaze was not focused, contrarily while listening to excerpt No. 30

a



b

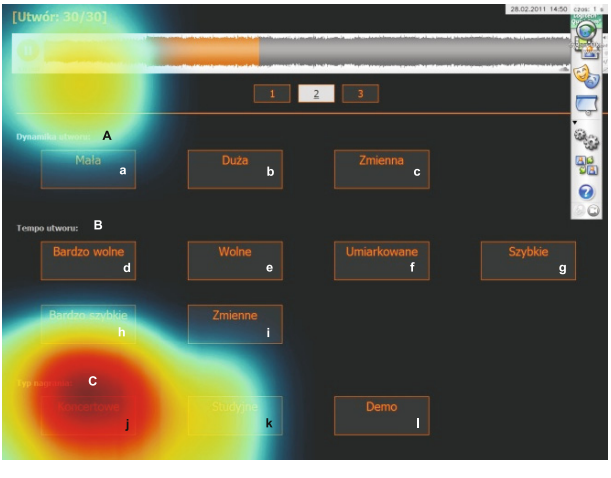
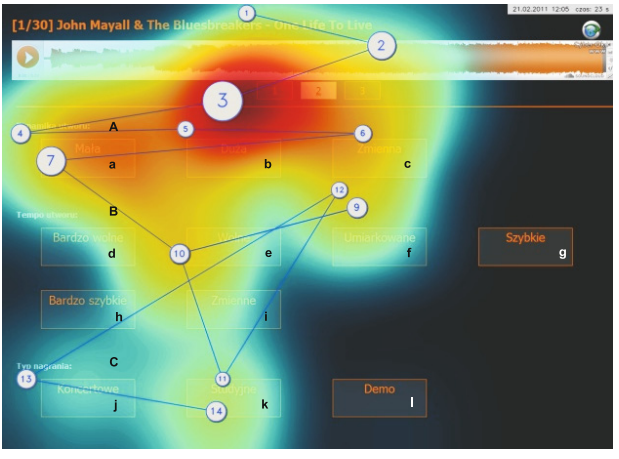


Fig. 17.6. Sample pages of the web-based form used in experiments (*in Polish*); denotations for Fig. 17.6a are as follows: D - music genre, E - type of music excerpt, m - jazz, n - blues, o - pop, p - rock, q - metal, r - classical, s - instrumental, t - vocal, u - instrumenta&vocal, v - chorale, w - orchestral and for Fig. 17.6b are as in Fig. 17.5; page No.1 - the element chosen corresponds to the type of the excerpt, that is, orchestral (left-bottom corner) (a), page No. 2 - the label indicated corresponds to the type of recording, that is, concert-live (left-bottom corner) (b)

the subject gaze was fixated on the target elements. Focusing on the target elements typically started after the fourth or fifth excerpt listened to. This happened with nearly all subjects.

a



b

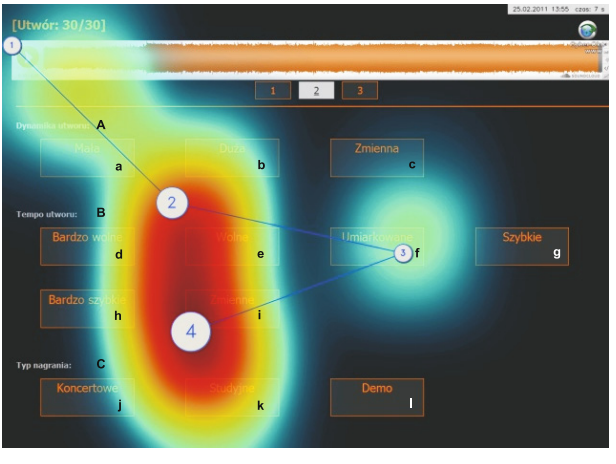


Fig. 17.7. An example of answers related to tempo tagging; both heatmaps and gaze plots are displayed; denotations are as in Fig. 17.5: 1st excerpt (a), 30th excerpt (b)

A more thorough analysis of tempo tagging was performed on one music excerpt. A musical piece that is referred to as *rap metal* or *funky metal* "*Know Your Enemy*" performed by the band *Rage Against The Machine* was randomly selected

for analysis. Subsequently, a few subjects of a given category were selected from Table 17.6 for further analysis. Three subjects were chosen from groups No. 1 and 3 each, while two other were chosen from group No. 4. Group No. 2 was omitted in the analysis (a single case).

Since the purpose of the experiment was to initially assess whether automatic tagging of music files by means of the gaze-tracking system was possible, this number of subjects seems to be sufficient to draw preliminary conclusions.

17.5.3.1 Result Analysis

First, it was assessed to what extent the choice of the musical piece tempo made by clicking a mouse depended on the gaze-tracking time. Although the analysis was not carried out on the basis of all collected data, it allowed to draw preliminary conclusions. However, the specificity of musical pieces in which the tempo is clearly changeable is noteworthy.

First, the choices of subjects who claimed not to listen to music or not to play any musical instrument were analyzed. The first subject selected the "Fast" tempo with a mouse, while his eyes stopped the longest at the "Very slow" button placed on the opposite side of the screen. Thus, these choices were completely different. On the other hand, the heatmap analysis also proved that the subject indicated the "Fast" button with the eyes. However, time of gazing on this element was much shorter. The second subject also selected the "Fast" tempo with a mouse. The choice recorded by the Cyber-eye system was different in this case as well, as it was the "Very fast" option. What is more, the time spent on looking at this element was three times longer than in the case of the clicked button. The heatmap, however, showed again that the subject's eyes first stopped on the mouse clicked button. In such a case, it would be reasonable to synchronize manual and system choices with the excerpts that were listened to. This, however, has not been planned at this stage.

Next, the choices of individuals who are not musicians, but who regularly listen to music, were verified. The results from three subjects were analyzed in this category. The first of them selected the "Moderate" tempo, while the Cyber-eye pointed to the "Fast" tempo next to the "Moderate" button. In this case the gaze-tracking time with regard to the clicked element was shorter than 0.1 s, but it was 0.8 s on the element which the gaze-tracking system indicated. The next subject chose the "Changeable" tempo and again the result did not fully correlate with that of the Cyber-eye which, on the basis of the longest gaze-tracking point, showed "Very slow". In the case of the last subject from this category and similarly to subjects who do not listen to music and are non-musicians, the "Fast" tempo was selected. In this case, the element observation time was shorter than 0.1s and was not taken into account by the Cyber-eye algorithm.

Finally, the data gathered from musicians that regularly listen to music (but of different genres) were analyzed. The first subject from this category made a correlated choice, that is, it was the same for clicking and Cyber-eye selection. The choice was the "Moderate" tempo. In this case, the selections with a mouse and the eye were

identical. It is noteworthy that such a confident selection may have resulted from the fact that it was the thirteenth track listened by the subject, so the factor of getting accustomed to the interface might have contributed to the outcome. It was the only case where the track that was listened to occurred so late. The second subject from this category showed the accurate tempo with a mouse, but the selection with the eyes was different (the heatmap, however, was again initially the same as the mouse selection). The last of the subjects selected the "Fast" option with a mouse, and the gaze stopped the longest at the "Slow" tempo. The heatmap analysis showed that the subject's eyes concentrated on the area around the elements of the form, rather than precisely inside the buttons.

Since the results are not fully satisfactory, the analysis of the data from the Cyber-eye considering fixation time must be accompanied by a more precise analysis of heatmaps and gaze plots. Additionally, these observations must be synchronized with the musical excerpt that was listened to, to know whether tempo was stable throughout the whole music excerpt. In many cases, the agreement of the choices was good, so it seems that in future analysis one should primarily focus on this aspect and less on the longest time of gaze tracking, especially in the case of the changeable tempo of a musical piece. Furthermore, the data analysis shows that the accuracy of the Cyber-eye may not be sufficient, as in many cases, it was possible to see that subject's eyes concentrated on the area of a given element, but not directly on it. It is also noteworthy that the temporal resolution may have been problematic when testing the correlation between a gaze-tracking point and a mouse selection.

As part of the preliminary analysis, it was observed that users/musicians achieved greater agreement between mouse and heat map selections of a musical piece tempo (a more intense red color means a longer gaze-tracking time with regard to a given element. It is also proved by the size of a circle with a digit in the gaze plots.

The results show, however, that in many cases the selection made with a mouse was different from that made by the user's eyes. It may partly depend on the construction of the prepared page. On the other hand, it seems that only the user's perfect preparation for this type of task and the agreement of their music tastes with the assessed musical piece may bring satisfactory results together with the lack of wrong answers in relation to the meta-description collected in the database. It may thus be concluded that the objectivization of the annotation process is necessary.

17.6 Conclusions

A review of issues related to music genre retrieval and classification was carried out in this chapter, Examples of social networking services that recommend music were provided. Furthermore, examples of experiments on music genre classification followed by an attempt to make the music annotation process more objective were presented. The created music database was tested for accuracy of classification performed in the WEKA and RSES environments. Selected classifiers, that is, SVM, decision trees, the Random Forest and the rough set-based algorithms under-

went optimized testing. The classification for the proposed feature vector revealed to be efficient in recognizing over 90% for the SVM method, 84% for the Random Forest method, 78% for the RS analysis, and around 77% for the J48 algorithm. The experiments performed also provided preliminary conclusions on the use of the gaze-tracking system in musical piece annotation, showing that such a system may be useful when verifying experts who support the process.

Acknowledgements. This research is funded by the project No. SP/I/1/77065/10 entitled: "Creation of universal, open, repository platform for hosting and communication of networked resources of knowledge for science, education and open society of knowledge", being a part of Strategic Research Programme "Interdisciplinary system of interactive scientific and technical information" funded by the National Center for Research and Development (NCBiR, Poland).

References

1. Aucouturier, J.-J., Pachet, F.: Representing musical genre: A state of art. Journal of New Music Research 32(1), 83–93 (2003)
2. Bazan, J.G., Szczuka, M.S.: The Rough Set Exploration System. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)
3. Benetos, E., Kotropoulos, C.: A tensor-based approach for automatic music genre classification. In: Proceedings European Signal Processing Conference, Lausanne, Switzerland (2008)
4. Bergstra, J., Casagrande, N., Erhan, D., Eck, D., Kegl, B.: Aggregate features and AdaBoost for music classification. Machine Learning 65(2-3), 473–484 (2006)
5. Bisesi, E., Parncutt, R.: An accent-based approach to automatic rendering of piano performance: preliminary auditory evaluation. Archives of Acoustics 36(2), 283–296 (2011)
6. Celma, O.: Music recommendation and discovery…in which Web? In: 122nd Audio Eng. Soc. Convention, Vienna (2007)
7. Eye Trcaking, Consumer Insights Group, http://www.consumerinsightsgroup.com/what-we-do/eye-tracking/
8. Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E.: Social media recommendation based on people and tags. In: Proc. SIGIR 2010, Geneva, Switzerland, July 19-23, pp. 194–201. ACM, New York (2010)
9. Holzapfel, A., Stylianou, Y.: Musical genre classification using nonnegative matrix factorization-based features. IEEE Transactions on Audio, Speech, and Language Processing 16(2), 424–434 (2008)
10. Hughes, D., Manaris, B., Zalonis, T.: Real-Time Similarity Retrieval of Music, Vocalizations, and Arbitrary Sound Recordings. In: Proc. Int. Symp. Music Information Retrieval ISMIR 2011 Conference, Miami, FL, USA (2011), http://ismir2011.ismir.net/latebreaking/LB-20.pdf
11. Hyoung-Gook, K., Moreau, N., Sikora, T.: MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval. Wiley & Sons (2005)
12. Głaczyński, J., Łukasik, E.: Automatic music summarization. A "thumbnail" approach. Archives of Acoustics 36(2), 297–309 (2011)

13. Kostek, B.: Soft Computing in Acoustics: Applications of neural networks, fuzzy logic and rough sets to musical acoustics. STUDFUZZ. Physica Verlag, New York (1999)
14. Kostek, B., Czyzewski, A.: Representing musical instrument sounds for their automatic classification. J. Audio Eng. Soc. 49, 768–785 (2001)
15. Kostek, B.: Perception-Based Data Processing in Acoustics. Applications to Music Information Retrieval and Psychophysiology of Hearing. Series on Cognitive Technologies. Springer, Heidelberg (2005)
16. Kostek, B., Kania, Ł.: Music information analysis and retrieval techniques. Archives of Acoustics 33(4), 483–496 (2008)
17. Kostek, B., Kupryjanow, A., Zwan, P., Jiang, W., Raś, Z.W., Wojnarski, M., Swietlicka, J.: Report of the ISMIS 2011 Contest: Music Information Retrieval. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS(LNAI), vol. 6804, pp. 715–724. Springer, Heidelberg (2011)
18. Kunka, B., Kostek, B., Kulesza, M., Szczuko, P., Czyzewski, A.: Gaze-tracking-based audiovisual correlation analysis employing quality of experience methodology. Intelligent Decision Technologies 32, 217–227 (2010)
19. Kunka, B., Kostek, B.: Exploiting audio-visual correlation by means of gaze tracking. International Journal of Computer Science and Applications 36, 104–123 (2010)
20. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. In: Proceedings 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, pp. 282–289 (2003)
21. Lidy, T., Rauber, A., Pertusa, A., Inesta, J.: Combining audio and symbolic descriptors for music classification from audio. In: Music Information Retrieval Information Exchange, MIREX (2007)
22. Lindsay, A., Herre, J.: MPEG-7 and MPEG-7 Audio – An Overview. JAES 49(7/8), 589–594 (2001)
23. Mandel, M., Ellis, D.: LABROSA's audio music similarity and classification submissions. In: Music Information Retrieval Information Exchange, MIREX (2007)
24. Ness, S., Theocharis, A., Tzanetakis, G., Martins, L.G.: Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs. In: 17 ACM International Conf. on Multimedia, Beijing, China, pp. 705–708. ACM, New York (2009)
25. Pampalk, E., Flexer, A., Widmer, G.: Improvements of audio-based music similarity and genre classification. In: Proc. Int. Symp. Music Information Retrieval, ISMIR, London, UK, pp. 628–633 (2005)
26. Pachet, F., Cazaly, D.: A taxonomy of musical genre. In: Proc. RIAO Content-Based Multimedia Information Access Conf. 2000, pp. 1238–1246. Centre des Hautes Etudes Internationales d'Informatique Documentaire - C.I.D, College de France (2003)
27. Panagakis, I., Benetos, E., Kotropoulos, C.: Music genre classification: a multilinear approach. In: Proc. Int. Symp. Music Information Retrieval, ISMIR, pp. 583–588. Drexel University in Philadelphia, PA (2008)
28. Pawlak, Z.: Rough sets. International J. Computer and Information Sciences 11, 341–356 (1982)
29. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets. LNCS. Springer, Heidelberg (2004-2010)
30. Symeonidis, P., Ruxanda, M.M., Nanopoulos, A., Manolopoulos, Y.: Ternary semantic analysis of social tags for personalized music recommendation. In: Proc. 9th Int. Symp. Music Information Retrieval, ISMIR, pp. 219–224. Drexel University in Philadelphia, PA (2008)
31. Schierz, A., Budka, M.: High–Performance Music Information Retrieval System for Song Genre Classification. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS(LNAI), vol. 6804, pp. 725–733. Springer, Heidelberg (2011)

32. Tzanetakis, G., Cook, P.: Musical genre classification of audio signal. IEEE Transactions on Speech and Audio Processing 10(3), 293–302 (2002)
33. Tzanetakis, G., Essl, G., Cook, P.: Automatic musical genre classification of audio signals. In: Proc. Int. Symp. Music Information Retrieval, ISMIR 2001, pp. 293–302 (2001); IEEE Transactions on Speech and Audio Processing
34. Wołkowicz, J., Kulka, Z., Kešelj, V.: N-gram-based approach to composer recognition. Archives of Acoustics 33(1), 43–55 (2008)
35. Wójcik, J., Kostek, B.: Computational complexity of the algorithm creating hypermetric rhythmic hypotheses. Archives of Acoustics 33(1), 57–63 (2008)
36. Wroblewski, J.: Finding minimal reducts using genetic algorithms. In: Wang, P.P. (ed.) Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences, JCIS 1995, September 28-October 1, pp. 186–189. Duke Univresity Durham, NC (1995)
37. Wroblewski, J.: Genetic algorithms in decomposition and classification problems. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, pp. 472–492. Physica-Verlag, Heidelberg (1998)
38. Żwan, P.: Automatic singing quality recognition employing artificial neural networks. Archives of Acoustics 33(1), 65–71 (2008)
39. Żwan, P., Kostek, B.: System for automatic singing voice recognition. J. Audio Eng. Soc. 56(9), 710–723 (2008)
40. ITU GazeGroup, Reserach on eye tracking and gaze interaction, http://www.gazegroup.org/research
41. Million Song Dataset, http://labrosa.ee.columbia.edu/millionsong/
42. The International Society for Music Information Retrieval/Intern. Conference on Music Information Retrieval website, http://www.ismir.net/
43. Rough-set Exploration System, http://logic.mimuw.edu.pl/"rses/RSESdoceng.pdf
44. WEKA classification system, http://www.cs.waikato.ac.nz/ml/weka/
45. http://en.wikipedia.org/wiki/ITunes
46. http://www.amazon.com/
47. http://www.emusic.com/
48. http://www.ilike.com/
49. http://www.jamendo.com/pl/
50. http://www.lala.com/
51. http://www.last.fm/
52. http://www.mufin.com/us/
53. http://musicovery.com/
54. http://www.pandora.com
55. http://www.slacker.com/

Chapter 18
Rough Support Vectors: Classification, Regression, Clustering

Pawan Lingras, Parag Bhalchandra, Cory Butz, and S. Asharaf

Abstract. Support vector techniques were proposed by Vapnik as an alternative to neural networks for solving non-linear problems. The concepts of margins in support vector techniques provides a natural relationship with the rough set theory. This chapter describes rough set theoretic extensions of support vector technologies for classification, prediction, and clustering. The theoretical formulations of rough support vector machines, rough support vector regression, and rough support vector clustering are supported with a summary of experimental results.

Keywords: Support vector machines, clustering, prediction, classification, rough sets, rough patterns, financial modeling, conservative and aggressive modeling, regression, ε-insensitive loss function.

18.1 Introduction

Support vector machines (SVMs) are based on the statistical learning theory, also called *Vapnik-Chervonenkis* (VC) theory, developed by Vapnik and colleagues [3, 7, 25, 26, 27, 28]. Classification based on SVM's extends single layer perceptrons for non-linearly separable data_sets with the help of kernel functions and optimization. The original proposal for classification was generalized to regression, followed by clustering. The concept of margin that was central to the support vector based classification is a natural conduit to its fusion with rough set theory. The margin can be

Pawan Lingras
Saint Mary's University Halifax, Canada

Pawan Lingras · Parag Bhalchandra
Swami Ramanand Teerth Marathwada University, Nanded, India

Cory Butz
University of Regina, Regina, Canada

S. Asharaf
Indian Institute of Management, Kozhikode, India

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 491–515.
springerlink.com

interpreted as the boundary region in rough set theory. Lingras and Butz [11, 12] described a rough set-based interpretation of support vector classification and proposed an efficient multi-classification extension based on rough set theory. A similar margin in support vector regression created with the help of ε-loss function led to the proposal for rough support vector regression by Lingras and Butz [13]. Ashraf et al. [1] extended the support vector clustering(SVC)[2] to propose a rough support vector clustering algorithm that can create irregular-shaped soft clusters with lower and upper bounds. This chapter provides a comprehensive introduction to the hybridization of rough set theory and support vector techniques for classification, regression, and clustering. Theoretical foundations are followed by algorithmic implementation as well as experimental results to provide reader with sufficient background to pursue further developments in rough support vector computing.

18.2 Rough Support Vector Machines for Multi-classification

Support vector machines (SVMs), proposed by Vapnik [25, 26, 28, 27], are used for creating functions from a set of labeled training data [21]. The function can be a classification function with binary outputs or it can be a general regression function. In this section, we will restrict ourselves to classification functions. For classification, SVMs operate by attempting to find a hypersurface in the space of possible inputs that splits the positive examples from the negative examples. The split will be chosen to have the largest distance from the hypersurface to the nearest of the positive and negative examples. Intuitively, this makes the classification correct for testing data that is near, but not identical, to the training data.

Let $\mathbf{x}$ be an input vector in the input space X. Let y be the output in $Y \in \{-1,1\}$. Let $S = \{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_i,y_i),\ldots\}$ be the training set used for supervised prediction. Let us define the inner product of two vectors $\mathbf{x}$ and $\mathbf{w}$ as

$$\langle \mathbf{x},\mathbf{w}\rangle \;=\; \sum_j x_j \times w_j, \tag{18.1}$$

where x_j and w_j are components of the vectors $\mathbf{x}$ and $\mathbf{w}$, respectively.

The linear separability restriction in perceptron is overcome by the use of a nonlinear transformation Φ as shown in Fig. 18.1(a). Here, we are using the class labels $\{1,2\}$ instead of $\{-1,1\}$, since we will be generalizing the binary classification problem to multi-classification.

The choice of the hyperplane in perceptron algorithm was arbitrary. SVMs use the size of margin between two classes to search for an optimal hyperplane, which bifurcates the maximum margin. Fig. 18.1(b) shows the concept of margin between two classes. The line with intercepts of b_1 and b_2 enclose the maximum margin between two classes. The line with intercept b is the optimal line separating the two classes. The problem of maximizing the margin can be reduced to an optimization problem [5,28]:

$$\text{Minimize } \langle \Phi(\mathbf{w}),\Phi(\mathbf{w})\rangle y \times \langle \Phi(\mathbf{x}),\Phi(\mathbf{w})\rangle \geq 0, \forall(\mathbf{x},y) \in S \tag{18.2}$$

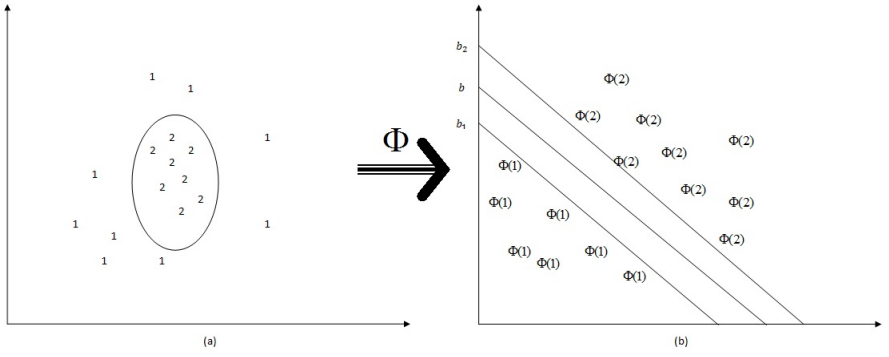


Fig. 18.1. Support Vector Machine Transformation to Linearly Separable Space

Usually, a high dimensional transformation is needed in order to obtain a reasonable prediction [30, 31]. Computational overhead can be reduced by not explicitly mapping the data to the feature space, but instead just working out the inner product in that space. In fact, SVMs use a kernel function K corresponding to the inner product in the transformed feature space as $K(\mathbf{x},\mathbf{w}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{w}) \rangle$.

Vapnik recognized the margin separating the two classes as an important issue in further theoretical developments. Lingras and Butz [12] described rough set-based interpretation of SVM binary classifications, where the margin between hyperplanes with intercept of b_1 and b_2 represent the boundary region between the two classes. Lingras and Butz described an algorithm for the calculation of b_1 and b_2 [12].

This rough set interpretation of SVM binary classification will allow us to create three equivalence classes, two equivalence classes for the lower bounds of each of the two classes, and third for the boundary region and define a rough set-based approximation space. This simple extension of an SVM classifier provides a basis for a more practical application, when the SVM transformation does not lead to a linear separable case. Cristianini [5] lists disadvantages of refining feature space to achieve linear separability. Often this will lead to high dimensions, which will significantly increase the computational requirements. Moreover, it is easy to overfit in high dimensional spaces, that is, regularities could be found in the training set that are accidental, which would not be found again in a test set. The soft margin classifiers [5] modify the optimization problem to allow for an error rate. In such a case, boundary region allows us to define an area of ambiguity. The error rate can also be used to determine the values of b_1 and b_2 in such a way that a large percentage of objects below the hyperplane given by b_1 belong to class 1, and an equally large percentage of objects below the hyperplane given by b_2 belong to class 2. This concept of soft margin was easily incorporated by Lingras and Butz in the algorithm for the calculation of b_1 and b_2. A rough set-based SVM binary classifier can then be defined by rules:

(R1) If $\langle \Phi(\mathbf{x}), \Phi(\mathbf{w}) \rangle + b_2 \geq 0$ then **x** belongs to class 2.
(R2) If $\langle \Phi(\mathbf{x}), \Phi(\mathbf{w}) \rangle + b_1 \leq 0$ then **x** belongs to class 1.
(R3) Otherwise, classification of **x** is uncertain

18.2.1 Extending Binary SVM's for Multi-classification

The SVM binary classification is extended to mulitclassification using two popular techniques, 1-v-1 and 1-v-r. The "one versus one" (1-v-1) strategy, also known as pair wise coupling, consists of constructing one SVM for each pair of classes. Thus, for a problem with n classes, $n \times (n-1)/2$ SVMs are trained to distinguish the samples of one class from the samples of another class. While doing so, the classification of an unknown pattern is done according to the maximum voting, where each SVM votes for one class. This makes 1-v-1 more accurate.

The "one versus rest" (1-v-r) strategy consists of constructing one SVM per class, which is trained to distinguish the samples of one class from the samples of all remaining classes. Usually, classification of an unknown pattern is done according to the maximum output among all SVMs. The total training time with the 1-v-1 strategy is larger than with the 1-v-r, because it is necessary to train more binary classifiers; but it is not necessarily true when the binary classifiers are SVMs. On the other hand, the individual binary classifiers in 1-v-1 are trained on a smaller sample set.

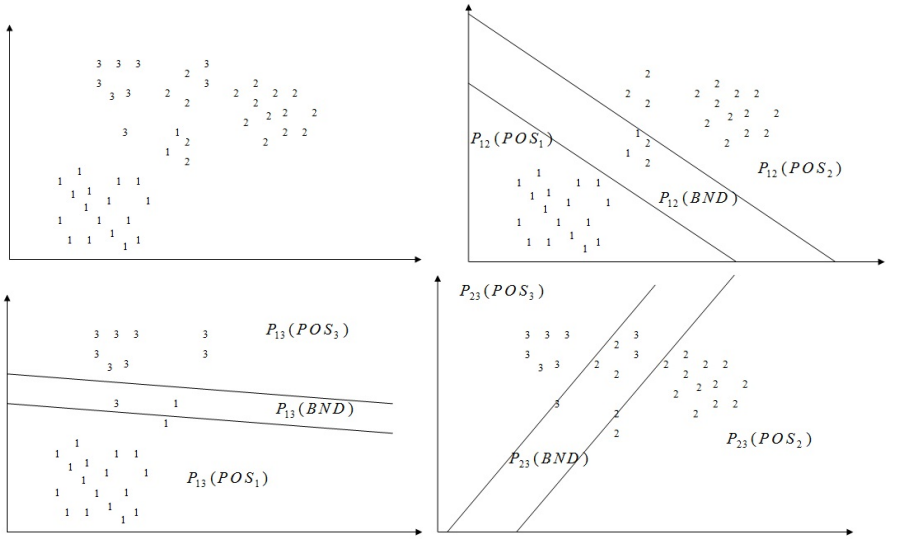


Fig. 18.2. Rough set-based 1-v-1 multi-classification

18.2.1.1 Rough Set-Based 1-v-1 Approach

Knerr et al. [8] suggested combining binary classifiers with an AND gate. While there are many other attempts to manage the large number of classifiers created by 1-v-1 approach, DAGSVM proposed by Platt et al. [22] is particularly interesting and popular. DAGSVM uses directed acyclic graphs to reduce the number of SVMs that need to be used during the testing and operational phase.

Lingras and Butz [12] provided a rough set interpretation of 1-v-1 classification that can lead to lower storage and lower computations in the operational phase. Fig. 18.2 depicts the approach for a three class problem. In order to understand the illustration, let us define some additional terms for pairwise classifications between class i and j in the 1-v-1 approach. For each pair, (i, j), that is separated by a binary SVM, we define three equivalence classes, let $P_{ij}(POS_i)$ and $P_{ij}(POS_j)$ be the lower bounds or positive regions of classes i and j, respectively. These two correspond to rules (R1) and (R2) Let $P_{ij}(BND)$ be the boundary region between the two classes corresponding to rule (R3). All the 1-v-1 classifiers can be combined to create lower bound and boundary regions for each class i as follows:

$$lower(class_i) = \bigcap_{j=1,\cdots,n,j\neq i} P_{ij}(POS_i) \tag{18.3}$$

$$upper(class_i) = \bigcup_{j=1,\cdots,n,j\neq i} (P_{ij}(BND) - lower(class_j)) \tag{18.4}$$

Detailed derivation of the above formulas can be found in [12].

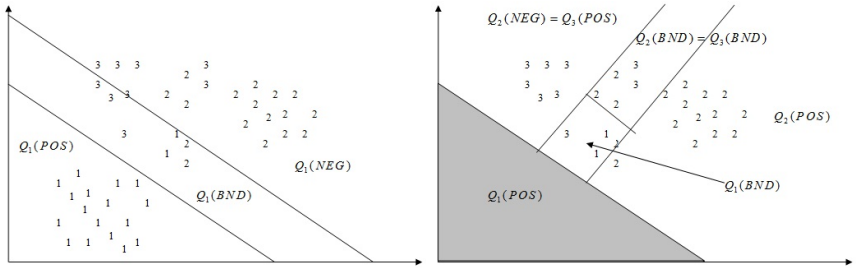
The rough set-based approach uses the same training time as the classical 1-v-1 or DAGSVM. It should perhaps be noted here that the upper bound given by eq. (18.4) will be polygonal, whose calculation needs additional computations. One of the advantages of rough set approach is that only two rules need to be stored for each class, one corresponding to the lower bound and another corresponding to the upper bound, as shown in the eq. (18.3) and eq. (18.4). Therefore, a total of $2 \times n$ rules are stored for the testing and operational phases, as opposed to $n \times (n-1)/2$ SVMs stored by 1-v-1 and DAGSVM approaches. Moreover, during the operational phase, the determination of membership of an object in a class will involve simply testing which lower bound the object belongs to. The time requirement for classification in the operational phase will be linear, that is, $O(n)$, the same as DAGSVM.

18.2.1.2 Rough Set-Based 1-v-r Approach

Without loss of generality, let us order the n classes such that class i contains at least as many objects as class $i+1$, where $1 \leq i < n$ in the training sample. Using the entire training sample and 1-v-r strategy, for $class_1$, create three equivalence classes $Q_1(POS)$ which is lower bound of $class_1$, $Q_1(NEG)$ is the lower bound of the collection of the rest of the classes, and $Q_1(BND)$ is the boundary region.

Since $Q_1(POS)$ definitely belongs to $class_1$, we can eliminate it from the next application of SVM classifier, that is, for the subsequent classes. For each subsequent class i, $1 < i < n$, refine $Q_{i-1}(BND) \cup Q_{i-1}(NEG)$ by creating $Q_i(POS)$, $Q_i(BND)$, $Q_i(NEG)$. For the last class, $Q_n(POS) = Q_{n-1}(NEG)$ and $Q_n(BND) = Q_{n-1}(BND)$. The upper and lower bounds for all n classes are defined as: $lower(class_i) = Q_i(POS)$ and $upper(class_i) = Q_i(POS) \cup Q_i(BND)$. As it is possible that some of the boundary regions may overlap with positive regions for subsequent classes, recalculate values of each of the boundary classes as:

$$Q_i(BND) = Q_i(BND) - \bigcup_{j=i}^{n} Q_j(POS). \tag{18.5}$$



(a) (b)

Fig. 18.3. Rough set-based 1-v-r multi-classification

Fig. 18.3 illustrates the rough-set based 1-v-r approach. For the $class_1$, $Q_1(POS)$, $Q_1(BND)$, $Q_1(NEG)$ are calculated using an SVM as shown in Fig. 18.3(a). As $Q_1(POS)$ only contains objects belonging to $class_1$, there is no need to further classify objects in $Q_1(POS)$. However, $Q_1(BND) \cup Q_1(NEG)$ should be further refined. Thus, for the next class ($class_2$), the resulting classification is shown in Fig. 18.3(b). Note that the shaded triangular area ($Q_1(POS)$) in Fig. 18.3(b) was eliminated from further classification, since it definitely belongs to $class_1$. 1-v-r classification allows us to identify the objects that definitely belong to $class_2$. In general case, the process will be further repeated until number of classes is reduced to two. In our example, we stop after applying the 1-v-r classification to class 2, since we have already classified $class_3$. That is, according to our algorithm, $Q_2(NEG) = Q_3(POS)$ and $Q_3(BND) = Q_2(BND)$. Fig. 18.3(b) shows the final classification.

An important feature of the rough set-based 1-v-r approach is that the sample size is reduced in each subsequent step. The reduction is optimized since the classes are ordered according to their cardinalities with the largest cardinality first. The lower bound (the positive region) of the largest class is eliminated from further

classification, followed by the next largest class, and so on. By reducing the size of training set, this elimination process may increase the training performance over the traditional 1-v-r approach. Lingras and Butz [12] provide detailed calculations on the extent of the reduction in training time.

18.2.2 Experiments with 1-v-1 and 1-v-r Approach

Lingras and Butz [12] demonstrated the implementation of rough set-based 1-v-1 and 1-v-r approaches for a synthetic data_set shown in Fig. 18.4 consisting of 150 objects. These 150 objects belong to three classes with each class containing 50 objects. As can be seen from Fig. 18.4, it is possible to separate most of the objects from these three classes using lines, but there will be a certain percentage of false positives and negatives that should belong to the boundary regions.

The experiments were carried out using Gist (http://svm.sdsc.edu/cgi-bin/nph-SVMsubmit.cgi), which provides values of discriminants, which can be used as surrogates of the distances from the hyperplane separating the positives from negatives. Usually, a positive discriminant corresponds to the object above the hyperplane, and negative discriminant indicates an object below the hyperplane. Therefore, we do

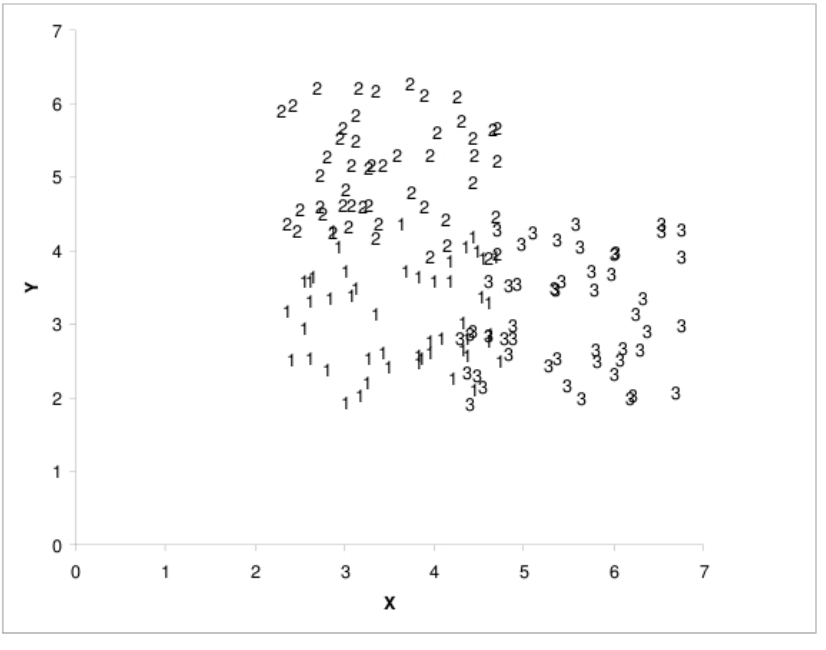


Fig. 18.4. Synthetic data for multi-classification

Table 18.1. Results of 1-v-1 classification

Region	Criteria	Cardinality
$P_{12}(POS_1)$	$0.2796 \leq d12$	35
$P_{12}(BND)$	$0.2796 > d12 > -0.3897$	32
$P_{12}(POS_2)$	$d12 \leq -0.3897$	33
$P_{13}(POS_1)$	$0.2767 \leq d13$	28
$P_{13}(BND)$	$0.2767 > d13 > -0.2105$	39
$P_{13}(POS_3)$	$d13 \leq -0.2105$	33
$P_{23}(POS_2)$	$0.083 \leq d23$	48
$P_{23}(BND)$	$0.083 > d23 > -0.2015$	3
$P_{23}(POS_3)$	$d23 \leq -0.2015$	49

not have to explicitly determine b_1 and b_2. More details about the use of discriminants can be found in [12].

18.2.2.1 Experimental Results Using 1-v-1

Let us use d12 to denote the discriminant for 1-v-1 classification of classes 1 and 2. Similarly, we also use d13 as discriminant for classification of classes 1 and 3, and d23 as discriminant for classification of classes 2 and 3. The results from application of rough set-based 1-v-1, shown in Table 18.1, can be combined using eq. 18.3 and eq. 18.4 to write the rules for upper and lower bounds of each class as follow:

- If $0.2796 \geq d12$ and $0.2767 \geq d13$ then, the object belongs to $lower(class_1)$
- If $d12 \leq 0.3897$ and $0.083 \geq d23$ then, the object belongs to $lower(class_2)$
- If $d13 \leq 0.2105$ and $d23 \leq 0.2015$ then, the object belongs to $lower(class_3)$

Similarly,

- If $d12 > 0.3897$ or $d13 > 0.2105$ then, the object belongs to $upper(class_1)$
- If $0.2796 < d12$ or $d23 > 0.2015$ then, the object belongs to $upper(class_2)$
- If $0.2767 < d13$ or $0.083 < d23$ then, the object belongs to $upper(class_3)$

As we can see, we only need to store six rules for the three classes. However, as mentioned before the rules can be complicated. Simplification and generalization of these rules is a promising research direction.

18.2.2.2 Experimental Results Using 1-v-1

Lingras and Butz [12] confirmed that 1-v-r classifications tend to produce less accurate results. Since all the classes had the same number of objects, we can choose any order for application of 1-v-r SVMs. We randomly choose the order $class_2, class_3$, $class_1$, The best results are found for $class_2$ as shown in Table 18.2, where d_{2r} is

Table 18.2. Results of 1-v-r classification

Region	Criteria	Cardinality
$Q_2(POS)$	$0.584 \leq d_{2r}$	33
$Q_2(BND)$	$0.584 > d_{2r} > -0.153$	32
$Q_2(NEG)$	$d_{2r} \leq -0.153$	85
$Q_3(POS)$	$0.3521 \leq d_{3r}$	28
$Q_3(BND)$	$0.3521 > d_{3r} > -0.2473$	44
$Q_3(NEG)$	$d_{3r} \leq -0.2473$	45

the discriminant of the 1-v-r classification to our synthetic data_set for $class_2$. Similarly, d_{3r} is the discriminant of the 1-v-r classification for $class_3$ after eliminating the positive region from $class_2$. We do not need to apply the 1-v-r classification to $class_1$, because the negative region of 1-v-r classification for $class_3$ can be used as the lower bound for $class_1$. Table 18.2 shows the regions, their criteria, and cardinalities which lead to the following rules:

- If $0.584 \geq d_{2r}$ then, the object belongs to $lower(class_2)$.
- If $0.3521 \geq d_{3r}$ then, the object belongs to $lower(class_3)$.
- If $d_{3r} < -0.2473$ then, the object belongs to $lower(class_1)$.

Similarly,

- If $d_{2r} > 0.153$ and $0.3521 > d_{3r} > -0.2473$ then, the object belongs to $upper(class2)$.
- If $0.584 < d_{2r}$ and $d_{3r} > 0.2473$ then, the object belongs to $upper(class3)$.
- If $0.584 < d_{2r}$ and $0.3521 < d_{3r}$ then, the object belongs to $upper(class_1)$.

It should be noted that the number of false positives and negatives in the lower bounds of all the classes are very high, compared with the 1-v-1 classification.

18.2.2.3 Semantics of Rough Set-Based Multi-classification

The rules corresponding to lower and upper bounds may provide better semantic interpretations of the multi-classification process than the other SVM approaches, which have been regarded as black-box models [23]. This is important, since da Rocha and Yager [23] advocate that describing the relationship between black-box approaches like SVMs with the logical rules approaches can lead to semantically enhanced network-based classifiers. It should be noted that the rules created by the application of Rough Set-based SVM cannot generally be written as a decision table, similar to conventional rough set approaches. However, the rules may still enable us to semantically analyze the classification process, especially if it were possible to associate semantic meaning to the discriminate values used in the rules developed in this section.

18.3 Dual Rough Support Vector Regression

Support vector regression (SVR) employs the margin concept for the regression problem with the help of ε-insensitive loss functions [24, 26]. SVR has been found especially useful in time series predictions [16, 17, 30, 31].

The following is a brief summary of SVR as described in detail by [30, 31]. We will use the same notations as SVM classifiers, except let $y \in \Re$. Furthermore, let $f(\mathbf{x})$ be a predicted value of y:

$$f(\mathbf{x}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{w}) \rangle + b. \tag{18.6}$$

The objective of SVR is to minimize the regression risk

$$R(f) = \frac{1}{2}\langle \Phi(\mathbf{w}), \Phi(\mathbf{w}) \rangle + C \sum_{i=1} l(f(\mathbf{x}_i), y_i), \tag{18.7}$$

where C is called the cost of error. The first term $\frac{1}{2}\langle \Phi(\mathbf{w}), \Phi(\mathbf{w}) \rangle$ can be seen as the margin in SVMs. The similarity between actual y and its prediction is given by the loss function $l(f(\mathbf{x}), y)$.

Vapnik [26] proposed an ε-insensitive loss function:

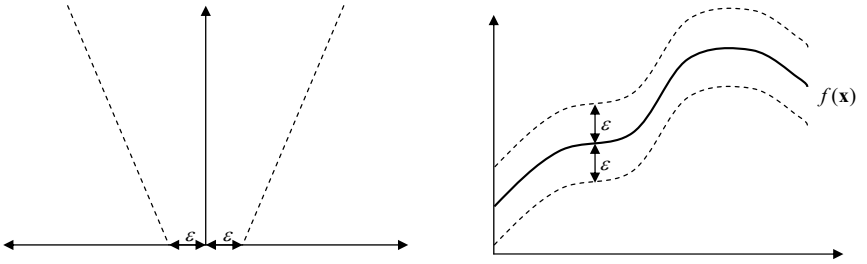
$$l_\varepsilon(f(\mathbf{x}), y) = \max(0, |y - f(\mathbf{x})| - \varepsilon) \tag{18.8}$$

shown in Fig. 18.5(a). The vertical axis denotes the loss. The horizontal axis corresponds to the value of $f(\mathbf{x})$. The two axes meet at $f(\mathbf{x}) = y$. If the predicted value is within $\pm\varepsilon$ of the actual value, the prediction is considered lossless. Fig. 18.5(b) shows how the actual values in the margin around the predicted function are considered acceptable or error-free. Increasing the ε value, reduces the number of support vectors. A large enough value of ε will lead to a constant regression function. The ε-insensitive loss function is ideally suited for modeling rough values as can be seen by the ε-tube around the prediction function in Fig. 18.5(b).

The corresponding SVR is called an ε-SVR. The minimization of Eq. (18.7) is reduced to a quadratic programming problem. The details of the formulation can be found in [30, 31]. Lingras and Butz [14] proposed dual extensions of SVR for modeling the rough patterns.

18.3.1 Rough Patterns

Pawlak proposed the concept of rough real functions which can be useful for rough controllers [19]. The notion of rough real functions was defined as an approximate value in case exact values cannot be represented by the given set of values. However, the notion can be used in a broader context. Lingras [9] used the rough values



(a) ε-insensitive loss function (b) Predictions with ε-SVR

Fig. 18.5. Support Vector Regression

to develop supervised and unsupervised neural networks [10, 20] and genetic algorithms [15]. This section describes rough values and patterns.

In some cases, a precise value of an entity may not be available. For example, one may estimate the current temperature to be between 20 and 25° C. In other cases, it may not even be possible to state a precise value. Many spatial (rainfall in Nova Scotia) or temporal (daily temperature) variables fall in this category. We cannot associate a precise value for daily temperature, only a range of value using the highest and lowest temperatures recorded on that day. We use rough or interval values to measure such quantities. For continuous variables, rough values are special cases of intervals as they focus on the end points. However, unlike intervals, rough values can also be used to represent a set of discrete values using the minimum and maximum values in the set. Let $Y = \{y_1, y_2, ..., y_n\}$ be a set of values collected for a variable such as daily temperature or stock market index. Each *rough value* y is denoted by a pair $(\underline{y}, \overline{y})$:

$$\underline{y} = \inf\{y \in Y\},$$

and

$$\overline{y} = \sup\{y \in Y\}.$$

Here, sup is defined as the maximum value from the set, while inf corresponds to the minimum value. The definitions of inf and sup can be modified to exclude the outliers. For example, one could use the bottom 5^{th} percentile value for $\underline{y}$ and top 5^{th} percentile value for $\overline{y}$. The above definition by Pawlak accommodates sets with continuous as well as discrete values. If the values are continuous, the set will be infinite and the resulting rough values correspond to the conventional notion of interval.

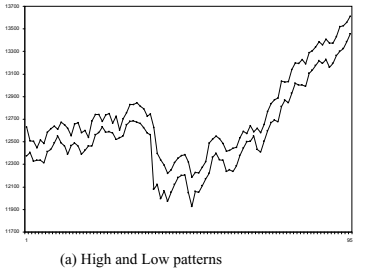
Rough patterns are sequences of rough or interval values [9]. We will look at a real world example of a rough pattern using a stock market index.

The *Dow Jones Industrial Average* (DJIA) is an index based on stock prices of the 30 most prominent companies listed on U.S. stock exchanges such as NYSE and NASDAQ. It is one of the most closely watched stock market indices in the world. The data used in this study was obtained from Yahoo! (www.yahoo.com). It

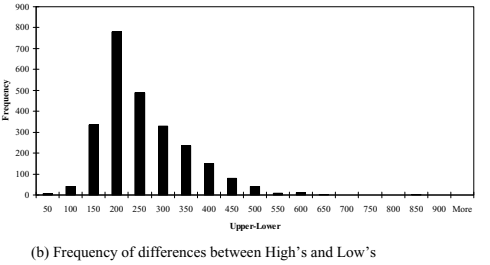
consisted of the date, opening value, closing value, high and low values, as well as the number of shares traded on the exchange. Data is only available for the trading days, that is, when the New York stock exchange was open. For example, in the ten years from May 21, 1997 to May 20, 2007, there were 2514 trading days.

Most of the prediction models are based on the closing values. The closing value of a stock or stock market index has impact on secondary investment instruments, such as mutual fund values and overseas markets. However, the traders on the New York stock exchange are reacting to minute-by-minute changes to stock prices, in addition to key indices like the DJIA. The stock exchanges are open from 10 a.m. to 3:30 p.m. from Monday through Friday with the exception of holidays. During these five and a half hours, one can get minute-by-minute updates on the values of DJIA. That will mean a total of 330 values per day. It will be difficult to manage such a large amount of data in any financial modeling. It is neither possible nor necessary to model/predict minute-by-minute values of the index. The traders, however, are interested in knowing how high or low a stock or index may go on a given day. For example, a trader who is looking to sell a stock or DJIA based financial derivative may wait until the high for the day is reached. Conversely, a trader who is looking to buy a stock or DJIA based financial derivative may wait until the low for the day is reached. Therefore, accurate prediction of trading range given by the rough pattern for a stock or stock index is an important part of stock market analysis.

Fig. 18.6(a) shows the rough pattern for the daily values of the DJIA from January 1 to May 20, 2007. The DJIA rough pattern consists of two curves. The top curve corresponds to the daily high's and the bottom one corresponds to the daily low values.



(a) High and Low patterns



(b) Frequency of differences between High's and Low's

Fig. 18.6. Dow Jones Industrial Average (DJIA) for early 2007

It is important to realize that there can be a considerable variation in the difference between high and low values, even though the general trend of the high and low values is essentially the same. Analysis of ten years of data from May 21, 1997 to May 20, 2007 shows the minimum difference to be 34.42, with a maximum value of 848.52. Fig. 18.6(b) shows the distribution of differences between highs and lows to be more or less normal. The average of the difference is 232, which is close to the

median and mode values. This analysis suggests that the high and low values should be separately analyzed.

18.3.2 Conservative and Aggressive Modeling of Rough Patterns

The formulation of rough ε-SVR to model rough patterns needs a definition of $\pm$ in the context of rough values. Let us assume that y is a rough value given by $(\underline{y},\overline{y})$. Lingras and Butz [14] described dual approaches termed as conservative and aggressive. In the conservative approach, the tube of predicted values will be tucked inside the tube of actual values. That is, the actual lower value $\underline{y}$ will not be higher than $\underline{f}(\mathbf{x})$ and the actual upper value $\overline{y}$ will not be lower than $\overline{f}(\mathbf{x})$. Let $fc(\mathbf{x}) = (\underline{fc}(\mathbf{x}),\overline{fc}(\mathbf{x}))$ be the conservative prediction of y such that

$$y = (\underline{fc}(\mathbf{x})-\varepsilon,\overline{fc}(\mathbf{x})+\varepsilon). \tag{18.9}$$

The aggressive model, on the other hand, will tell us how much lower the lower value can drop and how much higher the upper value can rise. The *aggressive* prediction, denoted fa, for a rough value $fa(\mathbf{x}) = (\underline{fa}(\mathbf{x}),\overline{fa}(\mathbf{x}))$ is more formally described as:

$$fa(\mathbf{x}) = (\underline{y}-\varepsilon,\overline{y}+\varepsilon). \tag{18.10}$$

Eq. (18.10) indicates that the actual lower value $\underline{y}$ will be equal or higher than $\underline{fa}(\mathbf{x})$ and the actual upper value $\overline{y}$ will be equal or lower than $\overline{fa}(\mathbf{x})$.

The conservative rough ε-insensitive loss function, denoted $lc_{r\varepsilon}(fc(\mathbf{x}),y)$, is defined as:

$$lc_{r\varepsilon}(fc(\mathbf{x}),y) = \underline{lc_{r\varepsilon}}(\underline{fc}(\mathbf{x}),\underline{y}) + \overline{lc_{r\varepsilon}}(\overline{fc}(\mathbf{x}),\overline{y}), \tag{18.11}$$

where $\underline{lc_{r\varepsilon}}(\underline{fc}(\mathbf{x}),\underline{y})$ is the lower component of the loss defined by:

$$\underline{lc_{r\varepsilon}}(\underline{fc}(\mathbf{x}),\underline{y}) = \begin{cases} \underline{y}-\underline{fc}(\mathbf{x}) & \text{if } \underline{y} \geq \underline{fc}(\mathbf{x}) \\ max(0,\underline{fc}(\mathbf{x})-\underline{y}-\varepsilon_d) & \text{otherwise} \end{cases}, \tag{18.12}$$

and $\overline{lc_{r\varepsilon}}(\overline{fc}(\mathbf{x}),\overline{y})$ is the upper component of the loss given by:

$$\overline{l_{r\varepsilon}}(\overline{fc}(\mathbf{x}),\overline{y}) = \begin{cases} \overline{fc}(\mathbf{x})-\overline{y} & \text{if } \overline{fc}(\mathbf{x}) \geq \overline{y} \\ max(0,\overline{y}-\overline{fc}(\mathbf{x})-\varepsilon_u) & \text{otherwise} \end{cases}. \tag{18.13}$$

The aggressive rough ε-insensitive loss function, denoted $la_{r\varepsilon}(fa(\mathbf{x}),y)$, is now defined as:

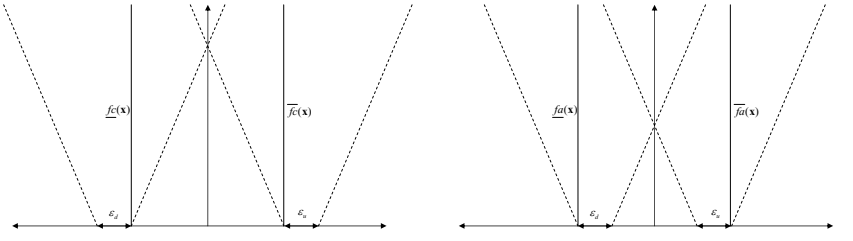
$$la_{r\varepsilon}(fa(\mathbf{x}),y) = \underline{la_{r\varepsilon}}(\underline{fa}(\mathbf{x}),\underline{y}) + \overline{la_{r\varepsilon}}(\overline{fa}(\mathbf{x}),\overline{y}), \tag{18.14}$$

where $\underline{la_{r\varepsilon}}(\underline{fa}(\mathbf{x}),\underline{y})$ is the lower component of the loss given by:

$$\underline{la_{r\varepsilon}}(\underline{fa}(\mathbf{x}),\underline{y}) = \begin{cases} max(0,\underline{fa}(\mathbf{x}) - \underline{y} - \varepsilon_d) & \text{if } \underline{y} \leq \underline{fa}(\mathbf{x}) \\ \underline{y} - \underline{fa}(\mathbf{x}) & \text{otherwise} \end{cases}, \tag{18.15}$$

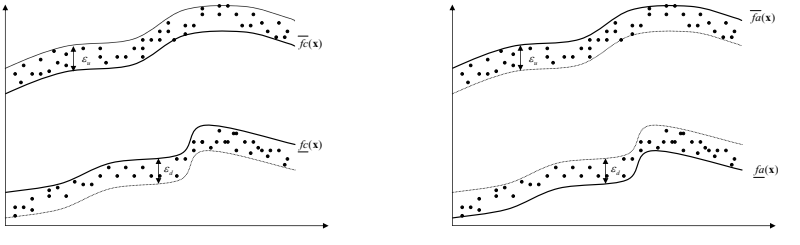
and $\overline{la_{r\varepsilon}}(\overline{fa}(\mathbf{x}),\overline{y})$ is the upper component of the loss defined as:

$$\overline{l_{r\varepsilon}}(\overline{fa}(\mathbf{x}),\overline{y}) = \begin{cases} max(0,\overline{y} - \overline{fa}(\mathbf{x}) - \varepsilon_u) & \text{if } \overline{fa}(\mathbf{x}) \leq \overline{y} \\ \overline{fa}(\mathbf{x}) - \overline{y} & \text{otherwise} \end{cases}. \tag{18.16}$$



(a) (b)

Fig. 18.7. Conservative and aggressive rough ε-insensitive loss functions



(a) (b)

Fig. 18.8. Conservative and aggressive modeling with dual rough ε-SVR

The conservative rough ε-insensitive loss function is shown in Fig. 18.7(a), which illustrates that we need to have an ε margin only on the outer side of the lower and upper prediction functions. The aggressive rough ε-insensitive loss function is

illustrated in Fig. 18.7(b). As can be seen, we need only to have an ε margin on the inner side of the lower and upper prediction functions.

The conservative modeling of rough patterns is depicted in Fig. 18.8(a). As can be seen, the predicted rough pattern will be inside the actual rough values of the variables as suggested by Eq. (18.9). Fig. 18.8(b) depicts the aggressive modeling of rough patterns. As suggested by Eq. (18.10), it can be observed that the predicted rough pattern will be outside the actual rough values of the variables.

The objective of conservative modeling is to construct a $fc(\mathbf{x})$ such that $lc_{r\varepsilon}$ is minimized. Aggressive modeling, on the other hand, attempts to minimize $la_{r\varepsilon}$. Lingras and Butz [14] describe the optimization process for both of these modeling approaches.

18.3.3 Empirical Analysis of Dual RSVR

The study data used in Lingras and Butz's [14] experiments consisted of the daily high and low values of the DJIA from May 21st to May 20th, 2007. There were a total of 250 trading days during the study period.

Conservative and aggressive RSVR described in the previous subsection were applied to model the DJIA rough pattern. The input vector consisted of the previous ten days high's and low's. Therefore, the model was applied to $250-10=240$ days. The output was the rough value for the next day.

Lingras and Butz [14] experimented with linear and polynomial Kernels and different values of $\varepsilon = 150, 75, 50$. The results seemed to significantly improve when ε was reduced from 150 to 75. The performance gain was not obvious when ε was further reduced to 50.

Error distribution for the two models is shown in Fig. 18.9. The error is calculated as $actual - predicted$. That means negative errors correspond to over-prediction and positive errors correspond to under-prediction. Fig. 18.9 (a) shows the frequency of errors using three types of bars for conservative modeling. The hollow bars for the lower prediction means that the actual value is less than 75 points (the value of ε_d) below the predicted value for the conservative model, and hence is acceptable according to the loss function $lc_{r\varepsilon}$. The striped bars for lower predictions mean that the values are over-predicted leading to a lower loss (because ε_d will be deducted for over-predictions of lower values). The solid bars indicate under-prediction of lower values. The reverse is true for upper predictions, that is, solid bars indicate over-prediction, striped bars are under-predictions leading to lower loss (ε_u will be deducted for under prediction of upper values), and hollow bars are under-predictions by less than 75 points (the value of ε_u) leading to zero loss. Based on Eq. (18.9) and the loss function $lc_{r\varepsilon}$, hollow bars are the most desirable, followed by striped bars, while the solid bars are least desirable.

Fig. 18.9 (b) also shows the frequency of errors using three types of bars for aggressive modeling. However, the meaning of the bars for the aggressive modeling is a mirror image of that for the conservative modeling. The hollow bars for the

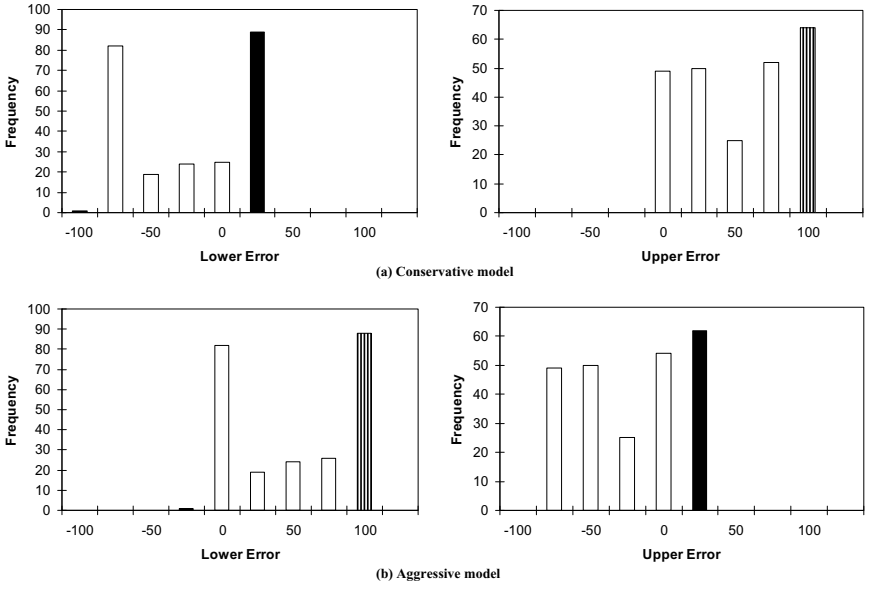


Fig. 18.9. Error distribution for conservative (top) and aggressive (bottom) rough ε-SVR modeling of rough DJIA pattern with $\varepsilon = 75$

lower prediction means that the actual value is less than 75 points (the value of ε_d) above the predicted value for the conservative model, and hence is acceptable according to the loss function $la_{r\varepsilon}$. The striped bars for lower predictions mean that the values are under-predicted leading to a lower loss (because ε_d will be deducted for under-predictions of lower values). The solid bars indicate over-prediction of lower values. The opposite is true for upper predictions, that is, solid bars indicate under-prediction, striped bars are over-predictions leading to lower loss (as ε_u will be deducted for over prediction of upper values), and hollow bars are over-predictions by less than 75 points (the value of ε_u) leading to zero loss. Similar to conservative modeling, based on Eq. (18.10) and the loss function $la_{r\varepsilon}$, the hollow bars are the most desirable, followed by striped bars, leaving solid bars serving as least desirable.

The abundance of hollow and striped bars, for both conservative and aggressive models, means that both approaches performed as expected. The errors for conservative modeling are on the outer sides of 0 (negative for lower values and positive for upper values), while they are on the inner side of 0 (positive for lower values and negative for upper values) for aggressive modeling. This observation clearly underscores the difference between the two philosphies. One can also notice a similarity between Fig. 18.9 (a) and the conservative loss function $lc_{r\varepsilon}$ given in Fig. 18.7(a). Similar correspondence can be drawn between Fig. 18.9 (b) and the aggressive loss function $la_{r\varepsilon}$ given in Fig. 18.7(b).

18.4 Rough Support Vector Clustering

Support vector clustering[2] is a clustering method that uses "kernel trick" [4]. Here, the computation in a high dimensional feature space is achieved using a Kernel function without explicitly mapping data points to the high dimensional feature space. In SVC, we look for the smallest sphere in a feature space that encloses the image of the data such as shown in Fig. 18.10. If this sphere is mapped back to data space, it forms a set of contours that enclose the data points, such as shown in Fig. 18.11. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour are associated with the same cluster. The kernel parameters can control the number of clusters. Here, the outliers are handled with the help of a soft margin formulation.

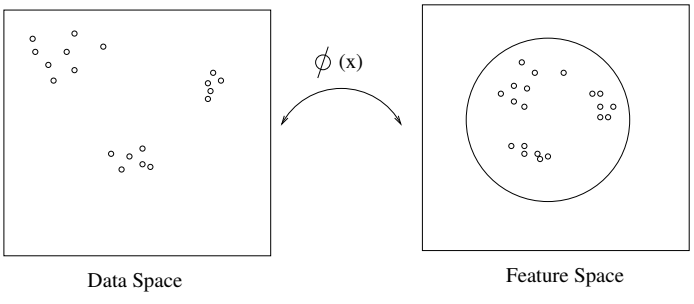


Fig. 18.10. Support Vector Clustering: data space to feature space mapping. Here, ϕ is the implicit non-linear transformation achieved by the kernel function.

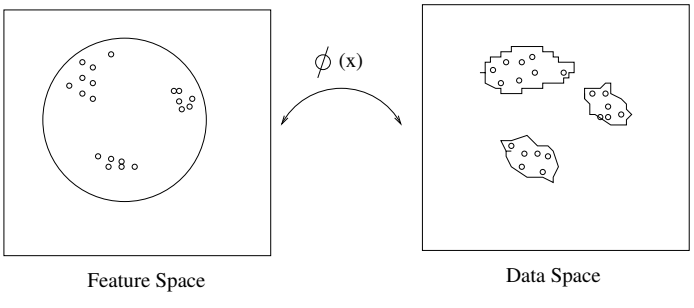


Fig. 18.11. Support Vector Clustering: the data space contours (clusters) obtained by the reverse mapping of the feature space sphere. Here, ϕ is the implicit non-linear transformation achieved by the kernel function.

To define the formulation, let $\{\mathbf{u}_i\} \subseteq U$ be an m-dimensional data set having n points, with $\mathbf{u}_i \in R^m$ being the data space. Now using a nonlinear transformation ϕ from U to some high dimensional feature space, we look for the smallest sphere of radius R enclosing all the points in U. Now the primal problem can be stated as:

$$\min_{R,\xi_i} R^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t.\ \ \|\phi(\mathbf{u}_i) - \mu\|^2 \leq R^2 + \xi_i, \qquad \xi_i \geq 0\ \forall i \tag{18.17}$$

Here, $C\sum_{i=1}^{n}\xi_i$ is the penalty term for the patterns with distance from the center of the sphere in feature space being greater than R (patterns that lie outside the feature space sphere), μ is the center of the sphere in the high dimensional feature space, and $\|.\|$ is the L_2 norm.

Since this is a Convex Quadratic Programming problem, it is easy to solve its Wolfe Dual [6] form. The dual formulation is:

$$\min_{\alpha_i} \sum_{i,j=1}^{n} \alpha_i\alpha_j K(\mathbf{u}_i,\mathbf{u}_j) - \sum_{i=1}^{n}\alpha_i K(\mathbf{u}_i,\mathbf{u}_i)$$

$$s.t.\ \ 0 \leq \alpha_i \leq C \quad for\ \ i = 1....n, \qquad \sum_{i=1}^{n}\alpha_i = 1 \tag{18.18}$$

Here, $K(\mathbf{u}_i,\mathbf{u}_j)$ represents the Kernel function giving the dot product $\phi(\mathbf{u}_i)\cdot\phi(\mathbf{u}_j)$ in the high dimensional feature space, and α_is are the Lagrangian multipliers.

The value of α_i decides whether a point $\phi(\mathbf{u}_i)$ is inside, outside, or on the sphere. The points with $0 < \alpha_i < C$ form the support vectors. Hence the radius of the sphere enclosing the image of the data points is given by: $R = G(\mathbf{u}_i)$, where $0 < \alpha_i < C$, and where

$$\begin{aligned} G^2(\mathbf{u}_i) &= \|\phi(\mathbf{u}_i) - \mu\|^2 \\ &= K(\mathbf{u}_i,\mathbf{u}_i) - 2\sum_{j=1}^{n}\alpha_j K(\mathbf{u}_j,\mathbf{u}_i) + \sum_{j,k=1}^{n}\alpha_j\alpha_k K(\mathbf{u}_j,\mathbf{u}_k). \end{aligned} \tag{18.19}$$

Now the contours that enclose the points in data space are defined by: $\{\mathbf{u} : G(\mathbf{u}) = R\}$. Thus, the computation in high dimensional feature space and also reverse mapping to find the contours in data space are avoided with the help of Kernel function. Once these contours are found, the objects are assigned to different clusters. The cluster assignments employ a geometric method involving $G(\mathbf{u})$, based on the observation that given a pair of points that belong to different clusters, any path that connects them must exit from the sphere in the feature space. So we can define an

adjacency matrix M by considering all pairs of points $\mathbf{u}_i$ and $\mathbf{u}_j$ whose images lie in or on the sphere in the feature space and then looking at the image of the path that connects them as:

$$M[i,j] = \begin{cases} 1 \text{ if } G(y) \leq R & \forall y \in [\mathbf{u}_i, \mathbf{u}_j] \\ 0 \text{ otherwise} \end{cases} . \tag{18.20}$$

Clusters are now defined as the connected components of the graph induced by M. The points that lie outside the sphere, known as bounded support vectors, can be assigned to the closest clusters.

Rough Support Vector Clustering (RSVC)[1, 29] is an extension of the SVC paradigm that employs rough set theory to achieve soft clustering. To discuss the method formally, let us use the notion of rough sets to introduce a Rough Sphere. A *Rough Sphere* is defined as a sphere having an inner radius R defining its lower approximation and an outer radius $T > R$ defining its upper approximation. As in SVC, RSVC also uses a kernel function to achieve computation in a high dimensional feature space. It tries to find the smallest rough sphere in the high dimensional feature space enclosing the images of all the points in the data_set. Now those points whose images lie within the lower approximation ($\underline{A}(\cdot)$) are points that definitely belong to exactly one cluster (called the hard core of a cluster) and those points whose images lie in the boundary region ($\overline{A}(\cdot) - \underline{A}(\cdot)$, that is, in the upper approximation $\overline{A}(\cdot)$ but not in the lower approximation $\underline{A}(\cdot)$) may be shared by more than one cluster (called the soft core of the cluster). Some points are permitted to lie outside the sphere and are termed outliers. By using a nonlinear transformation ϕ from data space to some high dimensional feature space, we seek the smallest enclosing rough sphere of inner radius R and outer radius T. Now the primal problem can be stated formally as:

$$\min_{R,T,\xi_i} \; R^2 + T^2 + \frac{1}{\upsilon n}\sum_{i=1}^{n}\xi_i + \frac{\delta}{\upsilon n}\sum_{i=1}^{n}\xi_i'$$

$$s.t \quad \|\phi(\mathbf{u}_i) - \mu\|^2 \leq R^2 + \xi_i + \xi_i' \quad 0 \leq \xi_i \leq T^2 - R^2 \quad \xi_i' \geq 0 \quad \forall i \tag{18.21}$$

Here, $\frac{1}{\upsilon n}\sum_{i=1}^{n}\xi_i$ is a penalty term for the patterns with distance from the center of the sphere in feature space being greater than R (patterns falling in the boundary region) and $\frac{\delta}{\upsilon m}\sum_{i=1}^{n}\xi'$ is a penalty term associated with the patterns whose distance from the center of the sphere in feature space is greater than T (patterns falling outside the rough sphere).

Since this is a Convex Quadratic Programming problem it is easy to write its Wolfe Dual. The Lagrangian can be written as:

$$L = R^2 + T^2 + \frac{1}{\upsilon n}\sum_{i=1}^{n}\xi_i + \frac{\delta}{\upsilon n}\sum_{i=1}^{n}\xi_i' + \sum_{i=1}^{n}\alpha_i(\|\phi(\mathbf{u}_i) - \mu\|^2 - R^2 - \xi_i - \xi_i')$$

$$-\sum_{i=1}^{n}\beta_i\xi_i+\sum_{i=1}^{n}\lambda_i(\xi_i-T^2+R^2)-\sum_{i=1}^{n}\eta_i\xi_i', \tag{18.22}$$

where the Lagrange multipliers $\alpha_i, \beta_i, \lambda_i$ and η_i are non-negative $\forall i$. Using the Karush-Kuhn-Tucker (KKT) [6] conditions on Eq. (18.22), we obtain:

$$\sum_{i=1}^{n}\alpha_i=2 \qquad \mu=\frac{1}{2}\sum_{i=1}^{n}\alpha_i\phi(\mathbf{u}_i)$$

$$\beta_i-\lambda_i=\frac{1}{\upsilon n}-\alpha_i \qquad \frac{\delta}{\upsilon n}-\alpha_i=\eta_i$$

$$\alpha_i(\|\phi(\mathbf{u}_i)-\mu\|^2-R^2-\xi_i-\xi_i')=0$$

$$\lambda_i(\xi_i-T^2+R^2)=0$$

$$\beta_i\xi_i=0 \qquad \eta_i\xi_i'=0$$

From the above equations the Wolfe Dual form can be written as:

$$\min_{\alpha_i}\sum_{i,j=1}^{n}\alpha_i\alpha_jK(\mathbf{u}_i,\mathbf{u}_j)-\sum_{i=1}^{n}\alpha_iK(\mathbf{u}_i,\mathbf{u}_i)$$

$$s.t \quad 0\le\alpha_i\le\frac{\delta}{\upsilon n} \quad for\ i=1....n, \qquad \sum_{i=1}^{n}\alpha_i=2. \tag{18.23}$$

Here, $K(\mathbf{u}_i,\mathbf{u}_j)$ represents the Kernel function giving the dot product $\phi(\mathbf{u}_i).\phi(\mathbf{u}_j)$ in the high dimensional feature space [2].

If $\delta>1$, we obtain RSVC. The formulation reduces to the original SVC for $\delta=1$. Also the values of α_i decide whether the pattern $\mathbf{u}_i$ falls in the lower approximation, the boundary region, or outside the feature space rough sphere. From KKT conditions on Eq. (18.22), it can be observed that image of points with

- $\alpha_i=0$ lie in the lower approximation.
- $0<\alpha_i<\frac{1}{\upsilon n}$ form the hard support vectors (support vectors marking the boundary of the lower approximation).
- $\alpha_i=\frac{1}{\upsilon n}$ lie in the boundary region (patterns that may be shared by more than one cluster).
- $\frac{1}{\upsilon n}<\alpha_i<\frac{\delta}{\upsilon n}$ form the soft support vectors (support vectors marking the boundary of the upper approximation).
- $\alpha_i=\frac{\delta}{\upsilon n}$ lie outside the rough sphere (bounded support vectors).

18.4.0.1 Cluster Assignment

Once the dual problem is solved to find the α_i values, the clusters can be obtained using the following strategy. Let us define:

$$R = G(\mathbf{u}_i) \quad : \quad 0 < \alpha_i < \frac{1}{\upsilon n}$$

$$T = G(\mathbf{u}_i) \quad : \quad \frac{1}{\upsilon n} < \alpha_i < \frac{\delta}{\upsilon n}, \tag{18.24}$$

where $G(\mathbf{u}_i)$ is defined in Eq. (18.19).

From the above equations we can define the contours that enclose the lower approximation of clusters in data space as: $\{\mathbf{u} : G(\mathbf{u}) = R\}$ and the contours that enclose the upper approximation of clusters in data space as: $\{\mathbf{u} : G(\mathbf{u}) = T\}$. Now the soft clusters in data space are found using a strategy similar to the one used in SVC. The *algorithm to find clusters* can now be given as follows.

- As in SVC, find the adjacency matrix M by considering all pairs of points $\mathbf{u}_i$ and $\mathbf{u}_j$ whose images in feature space either belong to the lower approximation of the rough sphere or are hard support vectors and then looking at the image of the path that connects them as

$$M[i,j] = \begin{cases} 1 \text{ if } G(y) \leq R & \forall y \in [\mathbf{u}_i, \mathbf{u}_j] \\ 0 \text{ otherwise} \end{cases}$$

- Find connected components for the graph represented by M. Each connected component found gives the lower approximation of a cluster $\mathbf{x_i}$.
- Find the boundary regions as:
  $\mathbf{u}_i \in \underline{A}(\mathbf{x_i})$ and pattern $\mathbf{u}_k \notin \underline{A}(\mathbf{x_j})$ for any cluster j,
  if $G(y) \leq T \ \forall y \in [\mathbf{u}_i, \mathbf{u}_k]$ then $\mathbf{u}_k \in (\overline{A}(\mathbf{x_i}) - \underline{\mathbf{A}}(\mathbf{x_i}))$

18.4.0.2 Role of υ and δ

From Eq. (18.23) it can be seen that the number of bounded support vectors is $n_{bsv} < 2\frac{\upsilon n}{\delta}$. For $\delta = 1$, $n_{bsv} < 2\upsilon n = \upsilon' n$ where $\upsilon' = 2\upsilon$. This corresponds to all the patterns $\mathbf{u}_i$ with $\|\phi(\mathbf{u}_i) - \mu\|^2 > R^2$. Since $\delta > 1$ for RSVC, we can say that $\frac{\upsilon'}{\delta}$ is the upper bound on the fraction of points permitted to lie outside T, and υ' is the upper bound on the fraction of points permitted to lie outside R. Hence, υ and δ together give us control over the width of boundary region and the number of bounded support vectors. Therefore, we can choose the values of υ and δ based on the percentage of the data we want to put in the soft core of the clusters, and what percentage we want to treat as outliers.

18.4.1 Experimental Results with RSVC

Experiments were done with a synthetically generated data set and a real world data set viz; Wine recognition data set [1], availabel at UCI Machine Learning repository from School of Information and Computer Sciences, University of California, Irvine.

18.4.1.1 Synthetic Data Set

The synthetic data is generated by sampling from four normal distributions with different mean values and uniform variance values. Thus the data set obtained get distributed in four clusters with some possible overlap between clusters. This synthetic data set and the clustering results obtained with $\upsilon = 0.25$ and $\delta = 1.25$ for RSVC are shown in Figure 18.12. From the clustering results, it may be observed that RSVC identified four clusters with some of the data points - shown by astericks - falls in the boundary region between the clusters as expected and some others does seem to belong to the upper approximation of any one cluster.

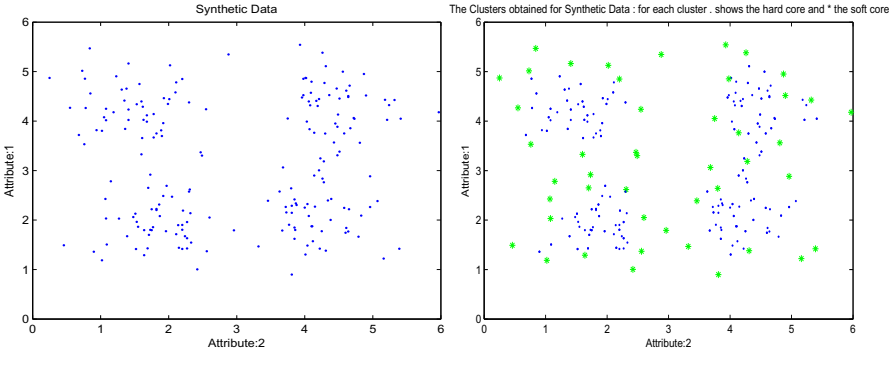


Fig. 18.12. Synthetic Data Set and RSVC result

18.4.1.2 Wine Recognition Data Set

The Wine recognition data set [1] is the results of a chemical analysis of three types (3 classes) of wines grown in the same region in Italy but derived from three different cultivators. This data is described by thirteen measurements (13 attributes) each representing the quantity of a constituent found in the wines. The data set consists of 59 instances of Classs 1 type of wine, 71 instances of Class 2 type of wine and 48 instances of Class 3 type of wine yielding a total of 178 patterns. Here we have

applied principal component analysis to reduce the dimensionality of the data set to two. The data set thus obtained and the result of applying RSVC with $\upsilon = 0.29$ and $\delta = 3.5$ on this data set are shown in Figure 18.13. As expected, the RSVC algorithm created three clusters roughly identifying the existing cluster sizes. Closer examination of Figure 18.13 reveals that there are a few objects - indicated by asterisks - that belong to the boundary region between these three clusters.

From the clustering results obtained from RSVC, it may be observed that there are some data points that belong to the upper approximation of only one cluster, but not shared by more than one clusters as demanded by the rough set theory [18]. Therefore, one can regard the clusters created by RSVC more as interval sets than as rough sets.

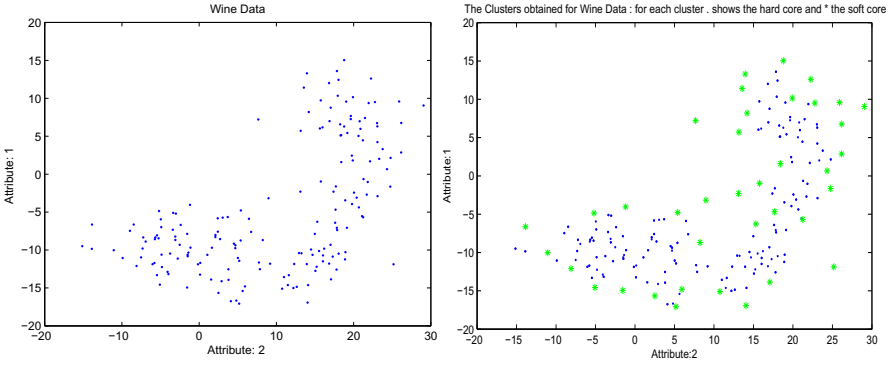


Fig. 18.13. Wine Data Set and RSVC result

18.5 Summary and Conclusions

There is a natural relationship between support vector techniques and the rough set theory. This chapter describes rough set theoretic extensions of SVM classification, regression, and clustering. The concepts of margin in SVM classification corresponds to the boundary regions in rough set theory. Use of boundary region is especially useful for soft margin SVM classifiers. Use of rough set view is shown to improve the performance of multi-classifications using both the 1-v-1 and 1-v-r approaches.

The ε-loss functions in support vector regression make it possible to create a band of lower and upper values for a function. The dual rough support vector regression described in this chapter modify the ε-loss functions using conservative and aggressive philosophies to model rough patterns in a financial time series.

Finally, the extension of support vector clustering using rough set theory allows us to create rough set representation of clusters with irregular boundaries, as opposed to traditional spherical surfaces in the conventional rough set clustering algorithms.

References

1. Asharaf, S., Shevade, S.K., Murty, N.M.: Rough support vector clustering. Pattern Recognition 38(10), 1779–1783 (2005)
2. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.N.: Support Vector Clustering. Journal of Machine Learning Research 2(2), 125–137 (2001)
3. Boser, B.E., Guyon, I., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proc. 5th Annual Workshop Computational Learning Theory, pp. 144–152. ACM, New York (1992)
4. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
5. Cristianini, N.: Support vector and kernel methods for pattern recognition (2003), http://www.support-vector.net/tutorial.html
6. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley-Interscience, New York (2000)
7. Hoffmann, A.: Vc learning theory and support vector machines (2003), http://www.cse.unsw.edu.au/~cs9444/Notes02/Achim-Week11.pdf
8. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: A stepwise procedure for building and training a neural network. In: Neurocomputing: Algorithms, Architectures and Applications. Springer, Heidelberg (1990)
9. Lingras, P.: Applications of rough patterns. In: Rough Sets in Data Mining and Knowledge Discovery, vol. 2, pp. 369–384. Springer (1998)
10. Lingras, P.: Fuzzy-rough and rough-fuzzy serial combinations in neurocomputing. Neurocomputing Journal 36, 29–44 (2001)
11. Lingras, P., Butz, C.J.: Interval set classifiers using support vector machines. In: Proc. the North American Fuzzy Inform Processing Socety Conference, pp. 707–710. Computer Society Press, Washington, DC (2004)
12. Lingras, P., Butz, C.J.: Rough set based 1-v-1 and 1-v-r approaches to support vector machine multi-classification. Information Sciences 177, 3298–3782 (2007)
13. Lingras, P., Butz, C.J.: Rough support vector regression. European Journal of Operational Research 206, 445–455 (2010)
14. Lingras, P., Butz, C.J.: Conservative and aggressive rough svr modeling. Theoretical Computer Science Journal Section on Theory of Natural Computing 412(42), 5885–5901(2011)
15. Lingras, P., Davies, C.: Applications of rough genetic algorithms. Computational Intelligence: An International Journal 17(3), 435–445 (2001)
16. Mukherjee, S., Osuna, E., Girosi, F.: Nonlinear prediction of chaotic time series using support vector machines. In: Principe, N.M.J., Giles, L., Wilson, E. (eds.) IEEE Workshop on Neural Networks for Signal Process, vol. VII, pp. 511–519. IEEE Computer Society Press, Washington, DC (1997)
17. Muller, K.R., Smola, A., Ratsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Predicting time series with support vector machines. In: ICANN, pp. 999–1004 (1997)
18. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
19. Pawlak, Z.: Rough real functions (1994), http://citeseer.ist.psu.edu/105864.html
20. Peters, J.F., Han, L., Ramanna, S.: Rough neural computing in signal analysis. Computational Intelligence 17(3), 493–513 (2001)
21. Platt, J.C.: Support vector machines (2003), http://research.microsoft.com/users/jplatt/svm.html

22. Platt, J.C., Cristianini, N., Shawe-Taylor, J.: Large margin dag's for multiclass classification. In: Advances in Neural Inform Processing Systems, pp. 547–553. MIT Press (2000)
23. da Rocha, A., Yager, R.: Neural nets and fuzzy logic. In: Kandel, A., Langholz, G. (eds.) Hybrid Architectures for Intelligent Systems, pp. 3–28. CRC Press, Ann Arbor (1992)
24. Smola, A., Scolkopf, B.: A tutorial on support vector regression. In: NeuroCOLT2 (1998)
25. Vapnik, V.: Estimation of Dependencies Based on Empirical Data. Nauka, Moscow (1979)
26. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, NY (1995)
27. Vapnik, V.: Statistical Learning Theory. Wiley, NY (1998)
28. Vapnik, V.N., Golowich, S., Smola, A.: Support vector method for function approximation, regression estimation and signal processing. In: Advances in Neural Information Processing Systems, vol. 9, pp. 281–287 (1997)
29. Varma, C.S., Asharaf, S., Murty, M.N.: Rough Core Vector Clustering. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 304–310. Springer, Heidelberg (2007)
30. Yang, H.: Margin Variations in Support Vector Regression for the Stock Market Prediction. M. Phil. dissertation, Department of Computer Science and Engineering, The Chinese University of Hong Kong (2003), http://www.svms.org/finance/Yang2003.pdf
31. Yang, H., Chan, L.-W., King, I.: Support Vector Machine Regression for Volatile Stock Market Prediction. In: Yin, H., Allinson, N., Freeman, R., Keane, J., Hubbard, S. (eds.) IDEAL 2002. LNCS, vol. 2412, pp. 391–396. Springer, Heidelberg (2002)

Chapter 19
Logic-Based Roughification

Linh Anh Nguyen and Andrzej Szałas

Abstract. The current chapter is devoted to *roughification*. In the most general setting, we intend the term *roughification* to refer to methods/techniques of constructing equivalence/similarity relations adequate for Pawlak-like approximations. Such techniques are fundamental in rough set theory. We propose and investigate novel roughification techniques. We show that using the proposed techniques one can often discern objects indiscernible by original similarity relations, what results in improving approximations. We also discuss applications of the proposed techniques in granulating relational databases and concept learning. The last application is particularly interesting, as it shows an approach to concept learning which is more general than approaches based solely on information and decision systems.

19.1 Introduction

Rough sets are typically used to model vague concepts and relationships. They are defined in terms of lower and upper approximations of crisp sets/relations, where approximations are in place when objects may be indiscernible due to incomplete, imprecise, and approximate data or knowledge. Indiscernibility of objects is modeled by similarity relations, originally assumed to be equivalence relations [19]. In general, the lower approximation of a set consists of objects whose similarity neighborhoods are contained in the set, while the upper approximation consists of objects whose similarity neighborhoods intersect the set. Similarity neighborhoods, often

Linh Anh Nguyen
Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
e-mail: nguyen@mimuw.edu.pl

Andrzej Szałas
Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland and
Dept. of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
e-mail: andrzej.szalas@mimuw.edu.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 517–543.
springerlink.com

being equivalence classes, are then substantial ingredients of approximate modeling and reasoning.[1]

The current chapter is devoted to *roughification*. In the most general setting, we intend that the term *roughification* refers to methods/techniques of constructing equivalence/similarity relations adequate for approximations. For example,

- in [19] as well as many later works (see, *e.g.*, [6, 20, 21] and references there), equivalence relations are constructed from information tables and decision tables, *for example*, by reducing the number of attributes
- in [27, 28],[2] equivalence relations are constructed by rough discretization and applied in clustering and classification
- in the light of [7], approximations can be constructed on the basis of logical theories, by projecting them into weaker languages.

We propose and investigate novel roughification techniques allowing one to construct suitable equivalence relations on the basis of background knowledge. We assume that knowledge is expressed by means of logical theories in the form of relational databases (*relational roughification*) and description logic theories (*terminological roughification*). The main idea depends on placing objects in the same equivalence class when they are indiscernible by a given logical theory. We show that using the proposed techniques one can often discern objects indiscernible by original similarity relations, so improve approximations. We also discuss applications of the proposed techniques in granulating relational databases and concept learning. The last application is particularly interesting, as it shows an approach to concept learning which is more general than approaches based solely on information and decision systems.

The first technique we propose is relational roughification, allowing one to obtain congruences on the basis of knowledge contained in relational databases. This technique, in fact, allows us to granulate arbitrary relational structures. We start with a simplified case, when such knowledge consists solely of similarity relations on objects, and show a natural technique (*similarity-based roughification*) allowing one to construct equivalence relations. This technique leads to better approximations than those offered by original similarities. As a general methodological outcome, we show that indiscernibility can actually be modeled by equivalence relations even if one initially uses weaker similarity relations, perhaps more intuitive in a given application domain. This places those other approaches back in the rough set context originally proposed and developed by Pawlak.

A more advanced version of roughification introduced in this chapter is based on bisimulations in the context of description logics. Namely, indiscernibility related to a given concept can be approximated using the largest auto-bisimulation with respect to the sublanguage consisting of concept names, role names and constructors the concept depends on. Such bisimulations are equivalence relations. We give

[1] For works, where similarity relations are not assumed to be equivalence classes, see [6, 8, 9, 12, 24, 29, 30] and references there.

[2] Where the term *roughification* has most probably been introduced in the context of discretization.

a logical account of this approach, investigate its theoretical properties and use it to study the problems of concept learning and concept approximation in information systems based on description logics.

Let us emphasize that all solutions we propose are tractable in the sense that data complexity of computing constructed new equivalence relations is in PTIME in the size of the underlying domain assuming that the underlying knowledge is given by means of relational/deductive databases or knowledge databases expressed by means of description logics.

The chapter is structured as follows. We start with basic definitions and preliminaries (Section 19.2). Then, in Section 19.3 we continue with similarity-based roughification and, in Section 19.4, with relational roughification. Section 19.5 is devoted to terminological roughification. Concluding remarks are contained in Section 19.6.

19.2 Preliminaries

Let Δ be a finite set, further called a *domain*. Elements of Δ are called *objects*. By a *relational structure* we understand a tuple $\langle \Delta, \{r_i\}_{i \in I} \rangle$, where for each $i \in I$, r_i is a relation on Δ.

For the sake of simplicity, in the chapter we consider one-sorted domains only. That is, we assume that objects are of the same type. The results we provide can be generalized in a straightforward manner to many-sorted structures. This, however, is not necessary for techniques we present.

A *signature* for relational structures consists of a finite set of *individual names* (i.e. *object names*), a finite set of *predicates* (i.e. relation names), and an arity mapping that associates each of the predicates with a natural number called the arity of the predicate.[3]

A relational structure over a signature Σ can be redefined to be a pair $I = \langle \Delta^I, \cdot^I \rangle$ consisting of a non-empty set Δ^I, called the *domain* of I, and a function $\cdot^I$, called the *interpretation function* of I, which maps each individual name a of Σ to an element a^I of Δ^I and maps each n-argument predicate p of Σ to an n-argument relation p^I on Δ^I.

By a *congruence* on $\langle \Delta, \{R_i\}_{i \in I} \rangle$ we understand any equivalence relation $\approx$ on Δ which *preserves* all relations $\{R_i\}_{i \in I}$, that is, such that for any $i \in I$, if R_i is an n-argument relation and $x_1 \approx x'_1, \ldots, x_n \approx x'_n$, then $R_i(x_1, \ldots, x_n) \equiv R_i(x'_1, \ldots, x'_n)$.

Let further $\sigma \subseteq \Delta \times \Delta$ be a binary relation on Δ, representing similarity on elements of Δ. It models indiscernibility on Δ in the sense that objects $x, x' \in \Delta$ are *indiscernible* whenever $\sigma(x, x')$ holds. The pair $\langle \Delta, \sigma \rangle$ is called a *similarity space*.

Given a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ and $A \subseteq \Delta$, Pawlak-like *approximations* of A w.r.t. $\mathcal{S}$ are defined as follows:

[3] For first-order logic, one would add to a signature also a finite set of function names and information about their arities but we concentrate on relations only.

- the *lower approximation* of A w.r.t. $\mathcal{S}$, denoted by $A_{\mathcal{S}^+}$ is defined by

$$A_{\mathcal{S}^+} \stackrel{\text{def}}{=} \{x \mid \forall y[\sigma(x,y) \rightarrow A(y)]\} \tag{19.1}$$

- the *upper approximation* of A w.r.t. $\mathcal{S}$, denoted by $A_{\mathcal{S}^\oplus}$ is defined by

$$A_{\mathcal{S}^\oplus} \stackrel{\text{def}}{=} \{x \mid \exists y[\sigma(x,y) \wedge A(y)]\}. \tag{19.2}$$

When $\mathcal{S}$ is known from context, we sometimes write $\underline{A}$ (respectively, $\overline{A}$) to denote the lower (respectively the upper) approximation of A w.r.t. $\mathcal{S}$, that is, $\underline{A} \stackrel{\text{def}}{=} A_{\mathcal{S}^+}$ and $\overline{A} \stackrel{\text{def}}{=} A_{\mathcal{S}^\oplus}$.

An information system in the rough set theory [19, 21, 20], called an *RS information system*, is usually defined to be a pair $\langle \Delta, Attrs \rangle$ of non-empty finite sets Δ and $Attrs$, where Δ is the *universe* of *objects*, and $Attrs$ is a set of *attributes*, that is, functions $A : \Delta \rightarrow V_A$, where V_A is the set of values of attribute A, called the *domain* of A.

19.3 Similarity-Based Roughification

Similarity-based roughification can be viewed as relational roughification introduced in Section 19.4. Namely, a relational structure can contain solely a similarity relation. However, similarities play a special role in defining relational roughifications. Also, intended applications make the technique interesting on its own. Therefore we discuss it separately.

19.3.1 Definitions

Observe that even if two objects x, x' are indiscernible w.r.t. a given similarity relation σ, that is, $\sigma(x,x')$ holds, it still might be the case that they can be discerned if there is an object x'' such that $\sigma(x,x'')$ and $\neg\sigma(x',x'')$. The same holds when there is an object x'' such that $\sigma(x'',x)$ and $\neg\sigma(x'',x')$. The first types of roughification reflect this phenomenon.

Given a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$, by a *forward similarity-based roughification* induced by $\mathcal{S}$ we understand relational structure $\mathcal{R}_{\mathcal{S}}^{\triangleright} = \langle \Delta, \rho_{\mathcal{S}}^{\triangleright} \rangle$, where:

$$\rho_{\mathcal{S}}^{\triangleright}(x,x') \stackrel{\text{def}}{\equiv} \forall x'' \big[\sigma(x,x'') \equiv \sigma(x',x'')\big]. \tag{19.3}$$

By a *backward similarity-based roughification* induced by $\mathcal{S}$ we understand relational structure $\mathcal{R}_{\mathcal{S}}^{\triangleleft} = \langle \Delta, \rho_{\mathcal{S}}^{\triangleleft} \rangle$, where:

$$\rho_{\mathcal{S}}^{\lhd}(x,x') \stackrel{\text{def}}{\equiv} \forall x''\big[\sigma(x'',x) \equiv \sigma(x'',x')\big]. \tag{19.4}$$

By a *similarity-based roughification* induced by $\mathcal{S}$ we understand relational structure $\mathcal{R}_{\mathcal{S}}^{\bowtie} = \langle \Delta, \rho_{\mathcal{S}}^{\bowtie}\rangle$, where

$$\rho_{\mathcal{S}}^{\bowtie} \stackrel{\text{def}}{=} \rho_{\mathcal{S}}^{\rhd} \cap \rho_{\mathcal{S}}^{\lhd}. \tag{19.5}$$

19.3.2 Properties

We have the following proposition.

Proposition 19.1. *Let $\mathcal{S} = \langle \Delta, \sigma\rangle$ be a similarity space. Then:*

- *$\rho_{\mathcal{S}}^{\rhd}$ and $\rho_{\mathcal{S}}^{\lhd}$ are equivalence relations on Δ*
- *$\rho_{\mathcal{S}}^{\bowtie}$ is a congruence on $\mathcal{S}$.*

Proof. The first claim is obvious by definitions (19.3) and (19.4).

To prove the second claim, note that $\rho_{\mathcal{S}}^{\bowtie}$ is the intersection of two equivalence relations, so it is an equivalence relation, too. To prove that it preserves σ, assume:

$$\rho_{\mathcal{S}}^{\bowtie}(x_1,x_1') \text{ and } \rho_{\mathcal{S}}^{\bowtie}(x_2,x_2'). \tag{19.6}$$

We have to show that $\sigma(x_1,x_2) \equiv \sigma(x_1',x_2')$. By (19.3)–(19.5) and (19.6), in particular we have:

$$\forall x''[\sigma(x_1,x'') \equiv \sigma(x_1',x'')] \text{ and } \forall y''[\sigma(y'',x_2) \equiv \sigma(y'',x_2')]. \tag{19.7}$$

Taking $x'' = x_2$ and $y'' = x_1'$ we have $\sigma(x_1,x_2) \equiv \sigma(x_1',x_2)$ and $\sigma(x_1',x_2) \equiv \sigma(x_1',x_2')$, so also $\sigma(x_1,x_2) \equiv \sigma(x_1',x_2')$. □

We also have the following proposition.

Proposition 19.2. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma\rangle$ with reflexive σ, we have that $\rho_{\mathcal{S}}^{\rhd} \subseteq \sigma$, $\rho_{\mathcal{S}}^{\lhd} \subseteq \sigma$ and $\rho_{\mathcal{S}}^{\bowtie} \subseteq \sigma$.*

Proof. Assume that $\rho_{\mathcal{S}}^{\rhd}(x,x')$. By (19.3), for all x'', $\sigma(x,x'') \equiv \sigma(x',x'')$. In particular, for $x'' = x'$ we have $\sigma(x,x') \equiv \sigma(x',x')$. By reflexivity of σ, we have that $\sigma(x',x')$ holds, so we also have that $\sigma(x,x')$ holds.

Analogously, using (19.4) we prove $\rho_{\mathcal{S}}^{\lhd} \subseteq \sigma$. Of course, $\rho_{\mathcal{S}}^{\bowtie} \subseteq \rho_{\mathcal{S}}^{\rhd}$, which proves the last inclusion. □

Observe that reflexivity of σ corresponds to the property that for any set A, $A_{\mathcal{S}^+} \subseteq A$ (see, *e.g.*, [10]). On the other hand, the weakest requirement placed on approximations, $A_{\mathcal{S}^+} \subseteq A_{\mathcal{S}^\oplus}$, is equivalent to the seriality of σ, that is, the property $\forall x \exists y[\sigma(x,y)]$. The following example shows that seriality is not sufficient to prove Proposition 19.2.

Example 19.1. Let $\mathcal{S} = \langle \{a,b,c\}, \sigma\rangle$, where $\sigma = \{(a,c),(b,c),(c,c)\}$. Obviously, σ is serial. On the other hand, $\rho_{\mathcal{S}}^{\rhd}(a,b)$ holds, while $\sigma(a,b)$ does not. □

Using Proposition 19.2 one can show that each $\mathcal{R}_{\mathcal{S}}^{\triangleright}$, $\mathcal{R}_{\mathcal{S}}^{\triangleleft}$ and $\mathcal{R}_{\mathcal{S}}^{\bowtie}$ approximates sets better than $\mathcal{S}$, as formulated in the following proposition.

Proposition 19.3. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ and any $A \subseteq \Delta$,*

$$A_{\mathcal{S}^+} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\triangleright})^+} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\bowtie})^+} \subseteq A \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\bowtie})^{\oplus}} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\triangleright})^{\oplus}} \subseteq A_{\mathcal{S}^{\oplus}}$$
$$A_{\mathcal{S}^+} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\triangleleft})^+} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\bowtie})^+} \subseteq A \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\bowtie})^{\oplus}} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{\triangleleft})^{\oplus}} \subseteq A_{\mathcal{S}^{\oplus}}.$$ □

19.3.3 Selected Applications

Proposition 19.3 shows that the similarity-based roughification may discern objects better than the original similarity relation. This allows us to sharpen perceptual capabilities, improving its accuracy. The following example illustrates this idea.

Example 19.2. Let a set of objects, say $\Delta = \{o_1, o_2, o_3\}$, be given. Assume that the accuracy of a sensor platform does not allow us to discern certain objects on the basis of their features. Such a situation is typically modeled by a similarity space $\langle \Delta, \sigma \rangle$ where, for example,

$$\sigma(o_1,o_1), \sigma(o_2,o_2), \sigma(o_3,o_3),$$
$$\sigma(o_1,o_2), \sigma(o_2,o_1), \sigma(o_2,o_3), \sigma(o_3,o_2),$$

that is, o_1 is indiscernible from itself and o_2, etc. On the other hand, one can discern o_1 and o_2 by comparing them with o_3. Such a comparison provides different results, allowing one to detect what object is being perceived. □

Similarity-based roughification can also be useful in decision rules mining. The obtained rules can be judged, among others, w.r.t. their classification accuracy. One faces here the overfitting/underfitting problem. Overfitting results in too many specialized rules, while underfitting causes poor classification results. The following example illustrates how can one tune decision rules using similarities resulting in better or worse approximations (by using Proposition 19.3).

Example 19.3. In the machine learning process one often obtains rules like:

$$\text{IF } bad\_condition(x) \text{ THEN } maintenance(x), \tag{19.8}$$

where objects are classified to be in "bad condition" on the basis of chosen attributes, say rust and moisture level. Particular examples in the training sets may be very specific. For example, an object o with rust level 0.743 and moisture level 0.92 may be marked as being in bad condition. One could then derive the following rule:

$$\text{IF } rust(x, 0.743) \text{ AND } moisture(x, 0.92) \text{ THEN } maintenance(x),$$

which definitely is too specific. One would also like to deduce that all *similar* (w.r.t. rust and moisture level) objects require maintenance, too:

$$\text{IF } \sigma(x,o) \text{ THEN } maintenance(x),$$

where σ is a similarity relation of a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with Δ consisting of value pairs $\langle rust, moisture \rangle$.

Now, if a given σ results in underfitting, one can use its roughification instead (or any suitable relation σ' such that $\rho_{\mathcal{S}}^{\bowtie} \subseteq \sigma' \subseteq \sigma$). Then, by moving σ' between the boundaries $\rho_{\mathcal{S}}^{\bowtie}$ and σ one can tune rules when new objects appear and are being classified. □

19.4 Relational Roughification

Relational roughification extends similarity-based roughification. Given a relational database, one can observe that object can be additionally discern by relations included in the database.

19.4.1 Definitions

Assume that a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ is given and $\mathcal{R}_{\mathcal{S}}^{\bowtie} = \langle \Delta, \rho_{\mathcal{S}}^{\bowtie} \rangle$ is the similarity-based roughification induced by $\mathcal{S}$.

Assume now that additional knowledge is provided by a relational or a deductive database. Even if two objects are indiscernible by $\rho_{\mathcal{S}}^{\bowtie}$, they may still be discernible by relations included in the database. For example, it might be the case that $\rho_{\mathcal{S}}^{\bowtie}(o,o')$ holds, while for a relation R in the database, it could be $R(\bar{a},o,\bar{b})$ and $\neg R(\bar{a},o',\bar{b})$. In such a case we can discern o and o' using R. We then have the following definition.

Given a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$, by a *relational roughification* induced by $\mathcal{S}$ and an m-argument relation R we understand relational structure $\mathcal{R}_{\mathcal{S}}^{R} = \langle \Delta, \rho_{\mathcal{S}}^{R} \rangle$, where:[4]

$$\rho_{\mathcal{S}}^{R} \stackrel{\text{def}}{=} \rho_{\mathcal{S}}^{\bowtie} - \{(x,x'),(x',x) \mid \exists x_1 \ldots \exists x_{m-1}[R(x_1,\ldots,x,\ldots,x_{m-1}) \wedge \neg R(x_1,\ldots,x',\ldots,x_{m-1})]\}. \tag{19.9}$$

Let us emphasize that in (19.9) we do not fix the position of x. For example, if R is a two-argument relation then (19.9) is to be understood as:

$$\rho_{\mathcal{S}}^{R} \stackrel{\text{def}}{=} \rho_{\mathcal{S}}^{\bowtie} - \big(\{(x,x'),(x',x) \mid \exists x_1[R(x_1,x) \wedge \neg R(x_1,x')]\} \cup \{(x,x'),(x',x) \mid \exists x_1[R(x,x_1) \wedge \neg R(x',x_1)]\}\big). \tag{19.10}$$

[4] Recall that $\rho_{\mathcal{S}}^{\bowtie}$ is the similarity-based roughification induced by $\mathcal{S}$ and defined by (19.5).

Observe that one can also consider tuples of relations rather than single relations. Namely, let $\{R_i\}_{i\in I}$ be a (finite) tuple of relations. Then:

$$\rho_{\mathcal{S}}^{\{R_i\}_{i\in I}} \stackrel{\text{def}}{\equiv} \bigcap_{i\in I} \rho_{\mathcal{S}}^{R_i}. \tag{19.11}$$

19.4.2 Properties

Let us first prove that the construction provided in the previous section indeed results in an equivalence relation.

Proposition 19.4. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space and R be a relation, $R \subseteq \Delta \times \ldots \times \Delta$. Then $\rho_{\mathcal{S}}^{R}$ is an equivalence relation on Δ.*

Proof. By Proposition 19.1, $\rho_{\mathcal{S}}^{\bowtie}$ is an equivalence relation.

Suppose $\rho_{\mathcal{S}}^{R}$ is not an equivalence relation. This could be caused by removing in (19.9) a pair (x,x') from $\rho_{\mathcal{S}}^{\bowtie}$. Let us now show that this cannot violate reflexivity, symmetry nor transitivity.

First note that reflexivity is preserved since there cannot exist $x_1,\ldots,x_{m-1}$ such that $R(x_1,\ldots,x,\ldots,x_{m-1})$ and, at the same time, $\neg R(x_1,\ldots,x,\ldots,x_{m-1})$.

Suppose now that $(x,x') \in \rho_{\mathcal{S}}^{R}$ and $(x',x) \notin \rho_{\mathcal{S}}^{R}$. This cannot happen since pairs (x,x') and $(x'x)$ are either not removed from $\rho_{\mathcal{S}}^{\bowtie}$ or are removed both. Therefore, symmetry is preserved.

Suppose that $(x,x'),(x',x'') \in \rho_{\mathcal{S}}^{R}$ and $(x,x'') \notin \rho_{\mathcal{S}}^{R}$. Since $\rho_{\mathcal{S}}^{R} \subseteq \rho_{\mathcal{S}}^{\bowtie}$, we have that $(x,x'),(x',x'') \in \rho_{\mathcal{S}}^{\bowtie}$, so also $(x,x'') \in \rho_{\mathcal{S}}^{\bowtie}$. Thus the assumption that $(x,x'') \notin \rho_{\mathcal{S}}^{R}$ implies that (x,x'') has been removed in (19.9), meaning that there are $x_1,\ldots,x_{m-1}$ such that

$$\begin{array}{ll} \text{either} & R(x_1,\ldots,x,\ldots,x_{m-1}) \wedge \neg R(x_1,\ldots,x'',\ldots,x_{m-1}) \\ \text{or} & \neg R(x_1,\ldots,x'',\ldots,x_{m-1}) \wedge R(x_1,\ldots,x'',\ldots,x_{m-1}). \end{array}$$

Consider the first case.[5] Since $R(x_1,\ldots,x,\ldots,x_{m-1})$ holds and $(x,x') \in \rho_{\mathcal{S}}^{R}$, we conclude that $R(x_1,\ldots,x',\ldots,x_{m-1})$ holds (otherwise the pair (x,x') was removed in (19.9)). Now from the fact that $R(x_1,\ldots,x',\ldots,x_{m-1})$ holds and assumption that $(x',x'') \in \rho_{\mathcal{S}}^{R}$, we also have that $R(x_1,\ldots,x'',\ldots,x_{m-1})$ and a contradiction is reached. □

The intersection of any collection of equivalence relations is also an equivalence relation. We then have the following corollary.

Corollary 19.1. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space and $\{R_i\}_{i\in I}$ be relations such that for all $i \in I$, $R_i \subseteq \Delta \times \ldots \times \Delta$. Then $\rho_{\mathcal{S}}^{\{R_i\}_{i\in I}}$ is an equivalence relation on Δ.* □

[5] The second case can be proved analogously.

Granular computing has been considered an important issue in rough set theory and applications [15, 23, 25, 26, 17, 22]. The following proposition shows that the constructed equivalence relation can serve as a basis for granulating relations.

Proposition 19.5. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space and R be a relation, $R \subseteq \Delta \times \ldots \times \Delta$. Then $\rho_{\mathcal{S}}^{R}$ is a congruence on $\langle \Delta, R \rangle$.*

Proof. By Proposition 19.4, $\rho_{\mathcal{S}}^{R}$ is an equivalence relation. To show that it preserves R, assume that:

$$\rho_{\mathcal{S}}^{R}(x_1, x_1'), \ldots, \rho_{\mathcal{S}}^{R}(x_m, x_m'). \tag{19.12}$$

We have to show that

$$R(x_1, \ldots, x_m) \equiv R(x_1', \ldots, x_m'). \tag{19.13}$$

To prove (19.13), we proceed by induction on $0 \leq k \leq m$:

$$R(x_1, \ldots, x_k, x_{k+1} \ldots, x_m) \equiv R(x_1', \ldots, x_k', x_{k+1} \ldots, x_m). \tag{19.14}$$

1. If $k = 0$ then (19.14) is obvious.
2. Assume that the theorem holds for $0 \leq k < m$. We shall show that it also holds for $(k+1)$:

$$\begin{array}{ll} R(x_1, \ldots, x_k, x_{k+1}, x_{k+2}, \ldots, x_m) \equiv & \text{(by inductive assumption (19.14))} \\ R(x_1', \ldots, x_k', x_{k+1}, x_{k+2}, \ldots, x_m) \equiv & \text{(by definition (19.9), assumption (19.12))} \\ R(x_1', \ldots, x_k', x_{k+1}', x_{k+2}, \ldots, x_m). & \end{array}$$

By analogy to Proposition 19.5 one can prove the following proposition providing a technique for granulating relational databases (see also Section 19.4.3).

Proposition 19.6. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space and $\langle \Delta, \{R_i\}_{i \in I} \rangle$ be a relational structure. Then $\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}}$ is a congruence on $\langle \Delta, R \rangle$.* □

By (19.9), we have that $\rho_{\mathcal{S}}^{R} \subseteq \rho_{\mathcal{S}}^{\bowtie}$. By Proposition 19.2 we then have the following proposition.

Proposition 19.7. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ and relation R on $\Delta \times \ldots \times \Delta$, we have that $\rho_{\mathcal{S}}^{R} \subseteq \sigma$.* □

As a consequence we have the following proposition.

Proposition 19.8. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ, any relation R on $\Delta \times \ldots \times \Delta$ and any $A \subseteq \Delta$,*

$$A_{\mathcal{S}^+} \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{R})^+} \subseteq A \subseteq A_{(\mathcal{R}_{\mathcal{S}}^{R})^{\oplus}} \subseteq A_{\mathcal{S}^{\oplus}}.$$ □

Remark 19.1. Note that relational roughification starts with some initial similarity relation and then improves its accuracy. If such a relation is not given, one can start with the total similarity relation $\sigma \stackrel{\text{def}}{=} \Delta \times \Delta$. If $\langle \Delta, \{R_i\}_{i\in I}\rangle$ is a relational structure then the resulting equivalence classes of $\rho_S^{\{R_i\}_{i\in I}}$ consist of objects indiscernible by relations $\{R_i\}_{i\in I}$. However, when $\sigma = \Delta \times \Delta$, similarity-based roughification provides no improvement, as in this case we have $\rho^{\bowtie} = \sigma$. □

19.4.3 Granulating Relational Databases

A relational database is a relational structure of the form $\langle \Delta, \{R_i\}_{i\in I}\rangle$ with finite Δ and I. Relational roughification allows us to granulate such databases in the sense that rather than using objects, we can use equivalence classes. Since an equivalence class may be represented by an arbitrary object it contains, such a granulation allows us to reduce the size of the database as well as consider classes of similar objects rather than singletons.

More precisely, given a relational database $DB = \langle \Delta, \{R_i\}_{i\in I}\rangle$ and a similarity space $\mathcal{S} = \langle \Delta, \sigma\rangle$, by a *granulation of DB w.r.t.* $\mathcal{S}$ we understand

$$DB/\rho_S^{\{R_i\}_{i\in I}} \stackrel{\text{def}}{=} \left\langle \Delta/\rho_S^{\{R_i\}_{i\in I}}, \{\mathbb{R}_i\}_{i\in I}\right\rangle, \tag{19.15}$$

where:

- $\Delta/\rho_S^{\{R_i\}_{i\in I}} \stackrel{\text{def}}{=} \{\|x\| \mid x \in \Delta\}$ is the set of equivalence classes of $\rho_S^{\{R_i\}_{i\in I}}$
- for $i \in I$, $\mathbb{R}_i(\|x_1\|, \ldots, \|x_m\|) \stackrel{\text{def}}{\equiv} R_i(x_1, \ldots, x_m)$.

By Proposition 19.6, $\mathbb{R}_i$ ($i \in I$) are well-defined.

Given a relational database $DB = \langle \Delta, \{R_i\}_{i\in I}\rangle$ and a similarity space $\mathcal{S} = \langle \Delta, \sigma\rangle$, rather than storing all tuples of relations in DB, it suffices to store tuples with representants of equivalence classes only. In addition, one needs to store $\rho_S^{\{R_i\}_{i\in I}}$ in the database, but the reduction od database size can be considerable.

19.5 Terminological Roughification

In this section we study roughification for information systems specified using the formalism of description logics (DLs). Such logics describe the domain of interest by means of individuals, concepts and roles [3, 4, 16]. A concept stands for a set of individuals, while a role stands for a binary relation between individuals. DLs are fragments of classical first-order logic and variants of modal logics. Indiscernibility in DLs is related to bisimulation.

In Sections 19.3 and 19.4 we had a particular similarity relation as a starting point for the construction of the final equivalence relation (but see Remark 19.1). Here we do not need such a starting relation. But whenever it is given, we can place it among roles.

19.5.1 Description Logics and Information Systems

A *DL-signature* is a set $\Sigma = \Sigma_I \cup \Sigma_C \cup \Sigma_R$, where Σ_I is a finite set of individual names, Σ_C is a finite set of *concept names*, and Σ_R is a finite set of *role names*. Concept names are unary predicates, while role names are binary predicates. We denote concept names by letters like A and B, role names by letters like r and s, and individual names by letters like a and b.

We will consider some (additional) *DL-features* denoted by I (*inverse*), O (*nominal*), Q (*quantified number restriction*), U (*universal role*), Self (*local reflexivity of a role*). A *set of DL-features* is a set consisting of some of these names.

Let Σ be a DL-signature and Φ be a set of DL-features. Let $\mathcal{L}$ stand for $\mathcal{ALC}_{reg}$, which is the name of a description logic corresponding to propositional dynamic logic (PDL). The DL language $\mathcal{L}_{\Sigma,\Phi}$ allows *roles* and *concepts* defined recursively as follows:

- if $r \in \Sigma_R$ then r is role of $\mathcal{L}_{\Sigma,\Phi}$
- if $A \in \Sigma_C$ then A is concept of $\mathcal{L}_{\Sigma,\Phi}$
- if R and S are roles of $\mathcal{L}_{\Sigma,\Phi}$ and C is a concept of $\mathcal{L}_{\Sigma,\Phi}$ then
  - ε, $R \circ S$, $R \sqcup S$, R^* and $C?$ are roles of $\mathcal{L}_{\Sigma,\Phi}$
  - $\top$, $\bot$, $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$ and $\exists R.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $I \in \Phi$ then R^- is a role of $\mathcal{L}_{\Sigma,\Phi}$
  - if $O \in \Phi$ and $a \in \Sigma_I$ then $\{a\}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$
  - if $Q \in \Phi$, $r \in \Sigma_R$ and n is a natural number then $\geq n\, r.C$ and $\leq n\, r.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $\{Q, I\} \subseteq \Phi$, $r \in \Sigma_R$ and n is a natural number then $\geq n\, r^-.C$ and $\leq n\, r^-.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $U \in \Phi$ then U is a role of $\mathcal{L}_{\Sigma,\Phi}$
  - if $\mathsf{Self} \in \Phi$ and $r \in \Sigma_R$ then $\exists r.\mathsf{Self}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$.

An *interpretation* in $\mathcal{L}_{\Sigma,\Phi}$ is a relational structure $I = \langle \Delta^I, \cdot^I \rangle$ over Σ. The interpretation function $\cdot^I$ is extended to complex roles and complex concepts as shown in Figure 19.1, where $\#\Gamma$ stands for the cardinality of the set Γ.

An (acyclic) *knowledge base* in $\mathcal{L}_{\Sigma,\Phi}$ is a pair $KB = \langle \mathcal{T}, \mathcal{A} \rangle$, where:

- $\mathcal{A}$ is a finite set, called the *ABox* of KB, consisting of *individual assertions* of the form $A(a)$ or $r(a,b)$, where $A \in \Sigma_C$, $r \in \Sigma_R$ and $a, b \in \Sigma_I$
- $\mathcal{T}$ is a finite list $(\varphi_1, \ldots, \varphi_n)$, called the *TBox* (terminological box) of KB, where each φ_i is a definition of one of the following forms:
  - $A = C$, where C is a concept of $\mathcal{L}_{\Sigma,\Phi}$ and $A \in \Sigma_C$ is a concept name not occurring in C, $\mathcal{A}$ and $\varphi_1, \ldots, \varphi_{i-1}$

$$
\begin{array}{rlrl}
(R \circ S)^I &= R^I \circ S^I & \top^I &= \Delta^I \\
(R \sqcup S)^I &= R^I \cup S^I & \bot^I &= \emptyset \\
(R^*)^I &= (R^I)^* & (\neg C)^I &= \Delta^I \setminus C^I \\
(C?)^I &= \{\langle x,x\rangle \mid C^I(x)\} & (C \sqcap D)^I &= C^I \cap D^I \\
\varepsilon^I &= \{\langle x,x\rangle \mid x \in \Delta^I\} & (C \sqcup D)^I &= C^I \cup D^I \\
U^I &= \Delta^I \times \Delta^I & \{a\}^I &= \{a^I\} \\
(R^-)^I &= (R^I)^{-1} & (\exists r.\mathsf{Self})^I &= \{x \in \Delta^I \mid r^I(x,x)\}
\end{array}
$$

$$
\begin{aligned}
(\forall R.C)^I &= \{x \in \Delta^I \mid \forall y\,[R^I(x,y) \text{ implies } C^I(y)]\} \\
(\exists R.C)^I &= \{x \in \Delta^I \mid \exists y\,[R^I(x,y) \text{ and } C^I(y)]\} \\
(\geq n R.C)^I &= \{x \in \Delta^I \mid \#\{y \mid R^I(x,y) \text{ and } C^I(y)\} \geq n\} \\
(\leq n R.C)^I &= \{x \in \Delta^I \mid \#\{y \mid R^I(x,y) \text{ and } C^I(y)\} \leq n\}
\end{aligned}
$$

Fig. 19.1. Interpretation of complex roles and complex concepts

- $r = R$, where R is a role of $\mathcal{L}_{\Sigma,\Phi}$ and $r \in \Sigma_R$ is a role name not occurring in R, $\mathcal{A}$ and $\varphi_1, \ldots, \varphi_{i-1}$.

The concept (respectively, role) names occurring in $\mathcal{A}$ are said to be *primitive* concepts (respectively, roles), while the concept (respectively, role) names occurring in the left hand side of '=' in the definitions from $\mathcal{T}$ are called *defined* concepts (respectively, roles).

An interpretation I in $\mathcal{L}_{\Sigma,\Phi}$ is a *model* of $KB = \langle \mathcal{T}, \mathcal{A}\rangle$ if

- for every assertion $A(a) \in \mathcal{A}$, we have $a^I \in A^I$
- for every assertion $r(a,b) \in \mathcal{A}$, we have $\langle a^I, b^I\rangle \in r^I$
- for every definition $(A = C) \in \mathcal{T}$, we have $A^I = C^I$
- for every definition $(r = R) \in \mathcal{T}$, we have $r^I = R^I$.

Example 19.4. Let

$$
\begin{aligned}
\Sigma_I &= \{Alice, Bob, Claudia, Dave, Eva, Frank, George\} \\
\Sigma_C &= \{Human, Female, Male, Adult, Man, Woman, \\
&\qquad Parent, ParentWMC, DecendantOfAlice\} \\
\Sigma_R &= \{has\_child, has\_descendant, has\_parent, has\_ancestor\} \\
\mathcal{A} &= \{Female(Alice), Female(Claudia), Female(Eva), Adult(Alice), \\
&\qquad Adult(Bob), Adult(Claudia), Adult(Dave), Adult(George), \\
&\qquad has\_child(Alice, Dave), has\_child(Bob, Dave), \\
&\qquad has\_child(Claudia, Eva), has\_child(Dave, Eva), \\
&\qquad has\_child(Claudia, Frank), has\_child(Dave, Frank)\}
\end{aligned}
$$

$$\begin{aligned}\mathcal{T} = (&Human = \top,\\ &Male = \neg Female,\\ &Woman = Human \sqcap Female \sqcap Adult,\\ &Man = Human \sqcap Male \sqcap Adult,\\ &Parent = \exists has\_child.\top,\\ &ParentWMC = (\geq 5\, has\_child.\top),\\ &has\_descendant = has\_child \circ has\_child^*,\\ &has\_parent = has\_child^\smile,\\ &has\_ancestor = has\_parent \circ has\_parent^*,\\ &DecendantOfAlice = \exists has\_ancestor.\{Alice\}).\end{aligned}$$

Then $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ is a knowledge base in $\mathcal{L}_{\Sigma,\Phi}$, with $\Phi = \{I, O, Q\}$. The definition $Human = \top$ states that the domain of any model of KB consists of human beings. Note that, *Female* and *Adult* are primitive concepts, and *has_child* is a primitive role of KB. □

A knowledge base as defined above is similar to stratified logic programs [1]. Hence, we define the *standard model* of a knowledge base $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{L}_{\Sigma,\Phi}$ to be the interpretation I such that:

- $\Delta^I = \Sigma_I$ (i.e. the domain of I consists of all the individual names of Σ)
- if A is a primitive concept of KB then $A^I = \{a \mid A(a) \in \mathcal{A}\}$
- if r is a primitive role of KB then $r^I = \{\langle a,b \rangle \mid r(a,b) \in \mathcal{A}\}$
- if $A \in \Sigma_C$ but A does not occur in KB then $A^I = \emptyset$
- if $r \in \Sigma_R$ but r does not occur in KB then $r^I = \emptyset$
- if $A = C$ is a definition from $\mathcal{T}$ then $A^I = C^I$
- if $r = R$ is a definition from $\mathcal{T}$ then $r^I = R^I$.

An *information system specified by a knowledge base* in $\mathcal{L}_{\Sigma,\Phi}$ is defined to be the standard model of the knowledge base in $\mathcal{L}_{\Sigma,\Phi}$. Note that such an information system is finite.

Example 19.5. Consider the knowledge base KB given in Example 19.4. The information system specified by KB is the interpretation I with:

$$\begin{aligned}\Delta^I &= \{Alice, Bob, Claudia, Dave, Eva, Frank, George\}\\ x^I &= x, \text{ for } x \in \{Alice, \ldots, George\}\\ Human^I &= \Delta^I\\ Female^I &= \{Alice, Claudia, Eva\}\end{aligned}$$

$$
\begin{aligned}
Male^I &= \{Bob, Dave, Frank, George\} \\
Adult^I &= \{Alice, Bob, Claudia, Dave, George\} \\
Woman^I &= \{Alice, Claudia\} \\
Man^I &= \{Bob, Dave, George\} \\
has\_child^I &= \{\langle Alice, Dave\rangle, \langle Bob, Dave\rangle, \\
&\qquad \langle Claudia, Eva\rangle, \langle Dave, Eva\rangle, \\
&\qquad \langle Claudia, Frank\rangle, \langle Dave, Frank\rangle\} \\
has\_parent^I &= (has\_child^I)^{-1} \\
has\_descendant^I &= has\_child^I \cup (has\_child^I \circ has\_child^I) \\
has\_ancestor^I &= (has\_descendant^I)^{-1} \\
Parent^I &= \{Alice, Bob, Claudia, Dave\} \\
ParentWMC^I &= \emptyset \\
DecendantOfAlice^I &= \{Dave, Eva, Frank\}.
\end{aligned}
$$

□

Observe that any RS information system with discrete (or Boolean) attributes can be represented as an information system in $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma_R = \emptyset$ and $\Phi = \emptyset$. Namely,

- if an attribute A of an RS information system is Boolean, that is, $V_A = \{\mathsf{true}, \mathsf{false}\}$, then it can be treated as a concept name, standing for the set $\{x \in \Delta \mid A(x) = \mathsf{true}\}$
- if A is a discrete attribute, with $V_A = \{v_1, \ldots, v_k\}$, then it can be replaced by concept names $A_{v_1}, \ldots, A_{v_k}$, where each A_{v_i} is interpreted as the set $\{x \in \Delta \mid A(x) = v_i\}$.[6]

Example 19.6. Let

$$
\begin{aligned}
Attrs &= \{Brand, Color, OpenOnSunday\} \\
V_{Brand} &= \{grocery, RTV\} \\
V_{Color} &= \{red, green, blue\} \\
\Delta &= \{shop_1, shop_2, shop_3, shop_4, shop_5\}
\end{aligned}
$$

and let attribute values of the objects be the following:

	Brand	*Color*	*OpenOnSunday*
$shop_1$	*RTV*	*red*	true
$shop_2$	*RTV*	*green*	true
$shop_3$	*RTV*	*blue*	true
$shop_4$	*grocery*	*red*	false
$shop_5$	*grocery*	*green*	false

[6] For example, if *Color* is an attribute with possible values *red*, *green* and *blue*, then we can replace it by concept names *Red*, *Green*, *Blue*, and instead of writing, *for example*, $Color(x) = red$, we can write $Red(x)$.

Then the RS information system $\langle \Delta, Attrs \rangle$ can be represented by the information system I in $\mathcal{L}_{\Sigma,\Phi}$ specified as follows:

$$\begin{aligned}
\Phi &= \emptyset \\
\Sigma_R &= \emptyset \\
\Sigma_I &= \{shop_1, shop_2, shop_3, shop_4, shop_5\} \\
\Sigma_C &= \{RTV, Grocery, Red, Green, Blue, OpenOnSunday\} \\
\Delta^I &= \Sigma_I \\
RTV^I &= \{shop_1, shop_2, shop_3\} \\
Grocery^I &= \{shop_4, shop_5\} \\
Red^I &= \{shop_1, shop_4\} \\
Green^I &= \{shop_2, shop_5\} \\
Blue^I &= \{shop_3\} \\
OpenOnSunday^I &= \{shop_1, shop_2, shop_3\}.
\end{aligned}$$

□

19.5.2 Bisimulation and Indiscernibility

In [5] Divroodi and Nguyen studied bisimulations for a number of DLs. In this subsection we generalize their notions and results to model indiscernibility of objects and study the problem of learning concepts. Let:

- Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$
- Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$
- I and I' be interpretations in $\mathcal{L}_{\Sigma,\Phi}$.

A binary relation $Z \subseteq \Delta^I \times \Delta^{I'}$ is called an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-*bisimulation* between I and I' if the following conditions hold for every $a \in \Sigma_I^\dagger$, $A \in \Sigma_C^\dagger$, $r \in \Sigma_R^\dagger$, $x, y \in \Delta^I$, $x', y' \in \Delta^{I'}$:

$$Z(a^I, a^{I'}) \tag{19.16}$$

$$Z(x,x') \Rightarrow [A^I(x) \Leftrightarrow A^{I'}(x')] \tag{19.17}$$

$$[Z(x,x') \wedge r^I(x,y)] \Rightarrow \exists y' \in \Delta^{I'}[Z(y,y') \wedge r^{I'}(x',y')] \tag{19.18}$$

$$[Z(x,x') \wedge r^{I'}(x',y')] \Rightarrow \exists y \in \Delta^I[Z(y,y') \wedge r^I(x,y)], \tag{19.19}$$

if $I \in \Phi^\dagger$ then

$$[Z(x,x') \wedge r^I(y,x)] \Rightarrow \exists y' \in \Delta^{I'}[Z(y,y') \wedge r^{I'}(y',x')] \tag{19.20}$$

$$[Z(x,x') \wedge r^{I'}(y',x')] \Rightarrow \exists y \in \Delta^I[Z(y,y') \wedge r^I(y,x)], \tag{19.21}$$

if $O \in \Phi^\dagger$ then

$$Z(x,x') \Rightarrow [x = a^I \Leftrightarrow x' = a^{I'}], \tag{19.22}$$

if $Q \in \Phi^\dagger$ then

$$\begin{array}{l}\text{if } Z(x,x') \text{ holds then, for every } r \in \Sigma_R^\dagger, \text{ there exists a bijection} \\ h : \{y \mid r^I(x,y)\} \to \{y' \mid r^{I'}(x',y')\} \text{ such that } h \subseteq Z,\end{array} \tag{19.23}$$

if $\{Q, I\} \subseteq \Phi^\dagger$ then (additionally)

$$\begin{array}{l}\text{if } Z(x,x') \text{ holds then, for every } r \in \Sigma_R^\dagger, \text{ there exists a bijection} \\ h : \{y \mid r^I(y,x)\} \to \{y' \mid r^{I'}(y',x')\} \text{ such that } h \subseteq Z,\end{array} \tag{19.24}$$

if $U \in \Phi^\dagger$ then

$$\forall x \in \Delta^I \, \exists x' \in \Delta^{I'} \, Z(x,x') \tag{19.25}$$

$$\forall x' \in \Delta^{I'} \, \exists x \in \Delta^I \, Z(x,x'), \tag{19.26}$$

if $\mathsf{Self} \in \Phi^\dagger$ then

$$Z(x,x') \Rightarrow [r^I(x,x) \Leftrightarrow r^{I'}(x',x')]. \tag{19.27}$$

A concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ is said to be *invariant for $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation* if, for every interpretations I and I' in $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma \supseteq \Sigma^\dagger$ and $\Phi \supseteq \Phi^\dagger$, and every $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation Z between I and I', if $Z(x,x')$ holds then $x \in C^I$ iff $x' \in C^{I'}$.

The following theorem can be proved in a similar way as [5, Theorem 3.4].

Theorem 19.1. *All concepts of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ are invariant for $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation.* □

An interpretation I is *finitely branching* (or *image-finite*) w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ if, for every $x \in \Delta^I$ and every $r \in \Sigma_R^\dagger$:

- the set $\{y \in \Delta^I \mid r^I(x,y)\}$ is finite
- if $I \in \Phi^\dagger$ then the set $\{y \in \Delta^I \mid r^I(y,x)\}$ is finite.

Let $x \in \Delta^I$ and $x' \in \Delta^{I'}$. We say that x is *$\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent* to x' if, for every concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, $x \in C^I$ iff $x' \in C^{I'}$.

The following theorem can be proved in a similar way as [5, Theorem 4.1].

Theorem 19.2 (The Hennessy-Milner Property). *Let Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$. Let I and I' be interpretations in $\mathcal{L}_{\Sigma,\Phi}$, finitely branching w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ and such that for every $a \in \Sigma_I^\dagger$, a^I is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to $a^{I'}$. Assume $U \notin \Phi^\dagger$ or $\Sigma_I^\dagger \neq \emptyset$. Then $x \in \Delta^I$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to $x' \in \Delta^{I'}$ iff there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation Z between I and I' such that $Z(x,x')$ holds.* □

We now have the following corollary.

Corollary 19.2. *Let Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, let Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and let I and I' be finite interpretations in $\mathcal{L}_{\Sigma,\Phi}$. Assume that $\Sigma_I^\dagger \neq \emptyset$ and, for every $a \in \Sigma_I^\dagger$, a^I is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to $a^{I'}$. Then the relation $\{\langle x,x'\rangle \in \Delta^I \times \Delta^{I'} \mid x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to $x'\}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and I'.* □

We say that I is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-*bisimilar* to I' if there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and I'. We say that $x \in \Delta^I$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-*bisimilar* to $x' \in \Delta^{I'}$ if there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and I' such that $Z(x,x')$ holds.

Remark 19.2. By Theorem 19.1, $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimilarity formalizes indiscernibility by the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. This is an important feature with many applications (see [7, 14, 31] for a more general context and numerous applications). Here let us emphasize that such indiscernibility relation provides the best approximations of a given concept expressed in the chosen sublanguage. Note that in [7, 14, 31] the underlying indiscernibility relation has not been constructed. □

An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and itself is called an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-*auto-bisimulation of* I. An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I is said to be the *largest* if it is larger than or equal to ($\supseteq$) any other $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I.

Given an interpretation I in $\mathcal{L}_{\Sigma,\Phi}$, by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ we denote the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I, and by $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$ we denote the binary relation on Δ^I with the property that $x \equiv_{\Sigma^\dagger,\Phi^\dagger,I} x'$ iff x is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to x'.

Theorem 19.3. *Let Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and I be an interpretation in $\mathcal{L}_{\Sigma,\Phi}$. Then:*

- *the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I exists and is an equivalence relation*
- *if I is finitely branching w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ then the relation $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$ is the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I (i.e. the relations $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$ and $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ coincide).*

□

Theorem 19.3 can be proved as [5, Proposition 5.1 and Theorem 5.2].

By *terminological roughification* we mean any technique that uses the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation relations as the equivalence relation for defining approximations.

The intended application areas are, in particular, concept learning and concept approximation in description logic-based information systems. Such applications and related techniques are studied in the next two subsections.

19.5.3 Concept Learning

Before presenting a method for learning concepts we first prove a theoretical result. We say that a set Y is *divided* by a set X if $Y \setminus X \neq \emptyset$ and $Y \cap X \neq \emptyset$. Thus, Y is not divided by X if either $Y \subseteq X$ or $Y \cap X = \emptyset$. A partition $P = \{Y_1, \ldots, Y_n\}$ is *consistent* with a set X if, for every $1 \leq i \leq n$, Y_i is not divided by X.

Theorem 19.4. *Let I be an information system in $\mathcal{L}_{\Sigma,\Phi}$, and let $X \subseteq \Delta^I$, $\Sigma^\dagger \subseteq \Sigma$ and $\Phi^\dagger \subseteq \Phi$. Then:*

1. *if there exists a concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $X = C^I$ then the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X*
2. *if the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X then there exists a concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $C^I = X$.*

Proof. As I is finite, it is finitely branching w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. By Theorem 19.3, $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ coincides with $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$.

Consider the first assertion and assume that $X = C^I$ for some concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. Since $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ coincides with $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$, if x and x' belong to the same equivalence class by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$, then x is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to x', and hence $x \in C^I$ iff $x' \in C^I$, that is, $\{x,x'\}$ is not divided by C^I. Therefore, the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X.

Consider the second assertion and assume that the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X. Let the partition be $\{Y_1,\ldots,Y_m\} \cup \{Z_1,\ldots,Z_n\}$, where $X = Y_1 \cup \ldots \cup Y_m$. Since Y_i and Z_j are different equivalence classes of $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$, we have that for each pair (i,j) with $1 \leq i \leq m$ and $1 \leq j \leq n$ there exists a concept $C_{i,j}$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $Y_i \subseteq C^I_{i,j}$ and $Z_j \cap C^I_{i,j} = \emptyset$. For each $1 \leq i \leq m$, let $C_i = C_{i,1} \sqcap \ldots \sqcap C_{i,n}$. Thus, $Y_i \subseteq C^I_i$, and $Z_j \cap C^I_i = \emptyset$ for all $1 \leq j \leq n$. Let $C = C_1 \sqcup \ldots \sqcup C_m$. Then, for all $1 \leq i \leq m$, $Y_i \subseteq C^I$, and for all $1 \leq j \leq n$, $Z_j \cap C^I = \emptyset$. Therefore, $C^I = X$. □

Let I be an information system in $\mathcal{L}_{\Sigma,\Phi}$, which can be either explicitly given as a finite interpretation in $\mathcal{L}_{\Sigma,\Phi}$ or specified by a knowledge base $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{L}_{\Sigma,\Phi}$. Let $A_d \in \Sigma_I$ be a concept name standing for the "decision attribute". In the case when I is specified by KB, assume that A_d is not defined by the TBox $\mathcal{T}$ of KB. Suppose that A_d can be expressed by a concept C in $\mathcal{L}_{\Sigma,\Phi}$ not using A_d, and I is given as a training information system. How can we learn that concept C on the basis of I? That is, how can we learn a definition of A_d on the basis of I?

On the basis of machine learning techniques one can suggest that A_d is definable in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, for some specific $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$. One can even guide the machine learning process by extending Σ, Φ and $\mathcal{T}$ with new concepts and new roles together with their definitions before suggesting $\Sigma^\dagger$ and $\Phi^\dagger$. Without such suggestions, one can take $\Sigma^\dagger = \Sigma$ or $\Phi^\dagger = \Phi$, or use some method to try different possible values of $\Sigma^\dagger$ and $\Phi^\dagger$.

In this subsection we assume that $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$ are given, and the task is to study a definition of A_d in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ on the basis of I.

Our idea for this problem is based on the following observation:

> if A_d is definable in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ then, by the first assertion of Theorem 19.4, A^I_d must be the union of some equivalence classes of Δ^I w.r.t. $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.

Our general method is as follows:

1. Starting from the partition $\{\Delta^I\}$, make subsequent granulations to reach the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.

- The granulation process can be stopped as soon as the current partition is consistent with A_d^I (or when some criteria are met).
- The task can be done in the spirit of [5, Algorithm 1] for the case $\Phi^\dagger \subseteq \{I, O, U\}$, which is based on Hopcroft's automaton minimization algorithm [13]. That algorithm of [5] runs in polynomial time and it can be extended to deal also with the other cases of $\Phi^\dagger$. Also, one can use another strategy, optimizing some measure related to "quality" of the generated partition, but not time complexity.
- In the granulation process, we denote the blocks created so far in all steps by $Y_1, \ldots, Y_n$, where the current partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ consists of only some of them. We do not use the same subscript to denote blocks of different contents (i.e., we always use new subscripts obtained by increasing n for new blocks). We take care that, for each $1 \leq i \leq n$:
  - Y_i is characterized by an appropriate concept C_i (such that $Y_i = C_i^I$)
  - we keep information about whether Y_i is divided by A_d^I
  - if $Y_i \subseteq A_d^I$ then $\mathit{LargestContainer}[i] := j$, where $1 \leq j \leq n$ is the subscript of the largest block Y_j such that $Y_i \subseteq Y_j \subseteq A_d^I$

2. At the end, let $j_1, \ldots, j_h$ be all the indices from $\{i_1, \ldots, i_k\}$ such that $Y_{j_t} \subseteq A_d^I$ for $1 \leq t \leq h$, and let $\{l_1, \ldots, l_p\} = \{\mathit{LargestContainer}[j_t] \mid 1 \leq t \leq h\}$. Let C be a simplified form of $C_{l_1} \sqcup \ldots \sqcup C_{l_p}$. Return C as the result.

Example 19.7. Consider the information system given in Example 19.5. Assume that we want to learn a definition of concept *Parent* in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger = \{\mathit{Adult}, \mathit{Female}, \mathit{has\_child}\}$ and $\Phi^\dagger = \emptyset$. The respective steps are:

1. $Y_1 := \Delta^I$, $\mathit{partition} := \{Y_1\}$
2. partitioning Y_1 by *Adult*:
   - $Y_2 := \{\mathit{Alice}, \mathit{Bob}, \mathit{Claudia}, \mathit{Dave}, \mathit{George}\}$, $C_2 := \mathit{Adult}$
   - $Y_3 := \{\mathit{Eva}, \mathit{Frank}\}$, $C_3 := \neg \mathit{Adult}$
   - $\mathit{partition} := \{Y_2, Y_3\}$
3. partitioning Y_2 by *Female*:
   - $Y_4 := \{\mathit{Alice}, \mathit{Claudia}\}$, $C_4 := C_2 \sqcap \mathit{Female}$
   - $\mathit{LargestContainer}[4] := 4$ (as $Y_4 \subseteq \mathit{Parent}^I$)
   - $Y_5 := \{\mathit{Bob}, \mathit{Dave}, \mathit{George}\}$, $C_5 := C_2 \sqcap \neg \mathit{Female}$
   - $\mathit{partition} := \{Y_3, Y_4, Y_5\}$
4. partitioning Y_3 by *Female*:
   - $Y_6 := \{\mathit{Eva}\}$, $C_6 := C_3 \sqcap \mathit{Female}$
   - $Y_7 := \{\mathit{Frank}\}$, $C_7 := C_3 \sqcap \neg \mathit{Female}$
   - $\mathit{partition} := \{Y_4, Y_5, Y_6, Y_7\}$
5. partitioning Y_4 by *has_child*:
   - $Y_8 := \{\mathit{Alice}\}$, $C_8 := C_4 \sqcap \exists \mathit{has\_child}.C_5$
   - $\mathit{LargestContainer}[8] := 4$

- $Y_9 := \{Claudia\},\ C_9 := C_4 \sqcap \neg\exists has\_child.C_5$
- $LargestContainer[9] := 4$
- $partition := \{Y_5, Y_6, Y_7, Y_8, Y_9\}$

6. partitioning Y_5 by *has_child*:

- $Y_{10} := \{Bob, Dave\},\ C_{10} := C_5 \sqcap \exists has\_child.\top$
- $LargestContainer[10] := 10$ (as $Y_{10} \subseteq Parent^I$)
- $Y_{11} := \{George\},\ C_{11} := C_5 \sqcap \neg\exists has\_child.\top$
- $partition := \{Y_6, Y_7, Y_8, Y_9, Y_{10}, Y_{11}\}$.

The obtained partition is consistent with $Parent^I$, with Y_8, Y_9, Y_{10} contained in $Parent^I$, and Y_6, Y_7, Y_{11} disjoint with $Parent^I$. (It is not yet the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.)

Since $LargestContainer[8] = LargestContainer[9] = 4$, the concept we take into account before simplification is $C_4 \sqcup C_{10}$, which is

$$(Adult \sqcap Female) \sqcup (Adult \sqcap \neg Female \sqcap \exists has\_child.\top).$$

This concept can be simplified to the following equivalent form

$$Adult \sqcap (Female \sqcup \exists has\_child.\top)$$

which does not match the intended definition $Parent = \exists has\_child.\top$. However, it is equivalent in I to an acceptable definition $Parent = Adult \sqcap \exists has\_child.\top$, as all women in I are parents. □

Example 19.8. Consider again the information system given in Example 19.5. Assume that we want to learn a concept definition of $X = \{Dave, Eva, Frank\}$ in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger = \{Alice, has\_child, has\_parent, has\_descendant, has\_ancestor\}$ and $\Phi^\dagger = \{O\}$. This task can be realized as follows:

1. $Y_1 := \Delta^I,\ partition := \{Y_1\}$
2. partitioning Y_1 by *Alice* using (19.22):

- $Y_2 := \{Alice\},\ C_2 := \{Alice\}$
- $Y_3 := \{Bob, Claudia, Dave, Eva, Frank, George\},\ C_3 := \neg\{Alice\}$
- $partition := \{Y_2, Y_3\}$

3. partitioning Y_3:

- The "selectors" are:
  - $\exists has\_child.C_3$, $\exists has\_parent.C_2$, $\exists has\_parent.C_3$,
  - $\exists has\_descendant.C_3$, $\exists has\_ancestor.C_2$, $\exists has\_ancestor.C_3$.
- If we apply the entropy gain measure then the best selectors are $\exists has\_parent.C_3$, $\exists has\_ancestor.C_2$, $\exists has\_ancestor.C_3$. Each of them partitions Y_3 into the following Y_4 and Y_5, but uses different C_4 and C_5:
  - $Y_4 := \{Dave, Eva, Frank\}$
  - $Y_5 := \{Bob, Claudia, George\}$.

4. Since the current partition $\{Y_2, Y_4, Y_5\}$ is consistent with X, the returned concept is C_4, which can be one of the following:
   - $\neg\{Alice\} \sqcap \exists has\_parent.\neg\{Alice\}$
   - $\neg\{Alice\} \sqcap \exists has\_ancestor.\{Alice\}$
   - $\neg\{Alice\} \sqcap \exists has\_ancestor.\neg\{Alice\}$.
5. If we test these solutions on the information system specified by the knowledge base that extends KB with the assertion $has\_child(Bob, George)$ then the solution $\neg\{Alice\} \sqcap \exists has\_ancestor.\{Alice\}$ has the best accuracy. □

Let us now describe our method in more details.

Let the current partition of Δ^I be $\{Y_{i_1}, \ldots, Y_{i_k}\}$. Consider partitioning of a block Y_{i_j} $(1 \leq j \leq k)$. We want to find a concept D of $\mathcal{L}_{\Sigma^\dagger, \Phi^\dagger}$, called a *selector*, to partition Y_{i_j}. Such a selector should actually partition Y_{i_j} into two non-empty parts (i.e. Y_{i_j} should be divided by D^I). It can be proved that to reach the partition corresponding to the equivalence relation $\sim_{\Sigma^\dagger, \Phi^\dagger, I}$ it suffices to consider the following kinds of selectors:

- A, where $A \in \Sigma_C^\dagger$: this is related to (19.17)
- $\exists r.C_{i_t}$, where $r \in \Sigma_R^\dagger$ and $1 \leq t \leq k$: this is related to (19.18) and (19.19)
- in the case $I \in \Phi^\dagger$:

  $\exists r^-.C_{i_t}$, where $r \in \Sigma_R^\dagger$ and $1 \leq t \leq k$: this is related to (19.20) and (19.21)
- in the case $O \in \Phi^\dagger$:

  $\{a\}$, where $a \in \Sigma_I^\dagger$: this is related to (19.22)
- in the case $Q \in \Phi^\dagger$:

  $\geq l\, r.C_{i_t}$ and $\leq m\, r.C_{i_t}$, where $r \in \Sigma_R^\dagger$, $1 \leq t \leq k$, $0 < l \leq \#C_{i_t}$ and $0 \leq m < \#C_{i_t}$: this is related to (19.23)
- in the case $\{Q, I\} \subseteq \Phi^\dagger$:

  $\geq l\, r^-.C_{i_t}$ and $\leq m\, r^-.C_{i_t}$, where $r \in \Sigma_R^\dagger$, $1 \leq t \leq k$, $0 < l \leq \#C_{i_t}$ and $0 \leq m < \#C_{i_t}$: this is related to (19.24)
- in the case $\mathsf{Self} \in \Phi^\dagger$:

  $\exists r.\mathsf{Self}$, where $r \in \Sigma_R^\dagger$: this is related to (19.27).

Note that the conditions (19.25) and (19.26) are always satisfied when $I' = I$ and Z is an equivalence relation.

In practice, we prefer as simple as possible definitions for the learnt concept. Therefore, it is worth to consider also the following kinds of selectors (despite that they are expressible by the above mentioned ones), where n is the largest block subscript used so far:

- $\exists r.C_i$, $\exists r.\top$ and $\forall r.C_i$, where $r \in \Sigma_R^\dagger$ and $1 \leq i \leq n$
- in the case $I \in \Phi^\dagger$: $\exists r^-.C_i$, $\exists r^-.\top$ and $\forall r^-.C_i$, where $r \in \Sigma_R^\dagger$ and $1 \leq i \leq n$

- in the case $Q \in \Phi^{\dagger}$: $\geq l\, r.C_i$ and $\leq m\, r.C_i$,
  where $r \in \Sigma^{\dagger}_R$, $1 \leq i \leq n$, $0 < l \leq \#C_i$ and $0 \leq m < \#C_i$
- in the case $\{Q, I\} \subseteq \Phi^{\dagger}$: $\geq l\, r^{-}.C_i$ and $\leq m\, r^{-}.C_i$,
  where $r \in \Sigma^{\dagger}_R$, $1 \leq i \leq n$, $0 < l \leq \#C_i$ and $0 \leq m < \#C_i$.

A concept C characterizing A_d in the training information system I may not match the intended meaning of A_d. In particular, all of the above mentioned kinds of selectors do not use role constructors (like $R \sqcup S$, $R \circ S$ or R^*). However, the user acquainted with the machine learning problem for A_d may extend Σ and the TBox of the knowledge base specifying I to define new complex roles and then choose an appropriate $\Sigma^{\dagger}$. One can explicitly consider also selectors that use complex roles. This latter approach, in our opinion, is not appropriate, as the search space will be too large.

We now describe partitioning the block Y_{i_j} using a selector D. Recall that Y_{i_j} should be divided by D^I. The partition is done as follows:

- $s := n+1,\ t := n+2,\ n := n+2,\ Y_s := C_{i_j} \sqcap D,\ Y_t := C_{i_j} \sqcap \neg D$
- If $Y_{i_j} \subseteq A^I_d$ then
  - $\mathit{LargestContainer}[s] := \mathit{LargestContainer}[i_j]$
  - $\mathit{LargestContainer}[t] := \mathit{LargestContainer}[i_j]$

  else if $Y_s \subseteq A^I_d$ then $\mathit{LargestContainer}[s] := s$
  else if $Y_t \subseteq A^I_d$ then $\mathit{LargestContainer}[t] := t$.
- The new partition of Δ^I becomes $\{Y_{i_1}, \ldots, Y_{i_k}\} \setminus \{Y_{i_j}\} \cup \{Y_s, Y_t\}$.

An important matter is: which block from the current partition should be partitioned first? which selector should be used to partition it? This affects both the "quality" of the final partition and time complexity of the process. Some guides and possible strategies are given below:

- If two selectors D and D' partition Y_{i_j} in the same way then the simpler one is "better". For example, if $D = \exists r.C_l$, $D' = \exists r.C_m$, $Y_m \subset Y_l$, and D, D' partition Y_{i_j} in the same way, then C_l is simpler than C_m and D is more preferred than D'. This technique together with the use of *LargestContainer* guarantees that one can continue granulating the partition without the risk of worsening the "quality" of the final result. (Remember, however, that different paths resulting in the same partition may give different results, with different "quality".)
- One may prefer to partition a block divided by A^I_d first. Partitioning such a block, we may use some measure to choose a selector. A possible way is to use the entropy gain measure. Among the blocks of the current partition that are divided by A^I_d, to choose a block to partition we can also use some measure. Once again, it may be the entropy gain measure, taking into account also the possible selectors.
- Note, however, that one may be able to partition a block divided by A^I_d only after a block not divided by A^I_d has been partitioned.

- Simplicity of selectors and concepts characterizing blocks should be taken into account (*e.g.*, by combining it with the entropy gain measure). Let's say the form A is simpler than $\exists r.B$ and $\{a\}$. One may put some limits on the number of nominals and the nesting depth of $\forall$ and $\exists$ in a concept characterizing a block.
- As a possible strategy, one may follow the idea of Hopcroft's automaton minimization algorithm. The hope is that reducing the total number of created blocks (in the whole granulation process) makes the concepts characterizing the blocks of the final partition simpler. Besides, apart from quality of the result, time complexity is also important.

As usual, we may also use backtracking to find different solutions. During the search, only the best choices are tried and we will keep only a bounded number of the best solutions (according to some measure). The final solution will be the one that has the best accuracy on a test information system.

Simplifying a concept C to obtain a final definition for A_d can be done as follows:

1. We first normalize C while preserving equivalence, *for example*, by using the method proposed in [18]. Such normalization uses negation normal form, which may be essential for cutoffs described below.
2. Given a test information system I', we then simplify the obtained concept, without preserving equivalence, by representing the concept as a tree and repeat the following operations until accuracy of the definition cannot be improved on I':
   - Cut off a leaf of the tree if it improves accuracy of the definition on I'.
   - If a subconcept of the definition can be replaced by a simpler one (*e.g.*, $\top$ or $\bot$) while not decreasing the accuracy on I' then do that replacement.
   - After each simplification, normalize the concept (preserving equivalence).

The other problems deserving consideration are: allowing a definition C not exactly matching A_d on I, and classifying a new object when inconsistencies occur. The first problem can be dealt with by using standard methods and some measures. Consider the second problem. Inconsistencies may occur as in the following situation: converting a training RS information system I_0 with a decision attribute *Color* and $V_{Color} = \{red, green, blue\}$ to a training information system I in DL with concepts *Red*, *Green*, *Blue* to be learnt, one may get concepts C_{red}, C_{green}, C_{blue} as the result of the learning process, which overlap on a real information system I''. A decision on whether an object x of I'' which belongs, *for example*, to both $C_{red}^{I''}$ and $C_{green}^{I''}$ should be classified as red or green can be made based on the accuracy of C_{red} and C_{green} on a test information system I'.

Note that an attempt to extend concept approximation using description logics was taken in [11] by using contextual indiscernibility relations used to represent uncertain concepts. A context is defined in [11] as a set of concepts. Roughly speaking, [11] proposes to define new atomic concepts by complex concepts and then to use those new atomic concepts for machine learning, applying traditional methods not based on description logics. The method we proposed is based on bisimulations and we find it much more promising for applications.

19.5.4 Bisimulation-Based Approximation of Concepts

The next problem we want to address is to learn a concept A_d not by giving its definition C (where A_d is a concept name and C is a complex concept), but by giving a pair $(\underline{C},\overline{C})$ of concepts, where $\underline{C}$ plays the role of a lower approximation of A_d and $\overline{C}$ plays the role of an upper approximation of A_d. This follows the lines of Pawlak's rough set theory.

The problem is specified as follows:

- **given:** a training information system I in $\mathcal{L}_{\Sigma,\Phi}$, a concept name $A_d \in \Sigma_C$, and a sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ of $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$
- **goal:** we want to learn an approximate definition of A_d, that is, a pair $(\underline{C},\overline{C})$ of concepts in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $\underline{C}^I \subseteq A_d^I \subseteq \overline{C}^I$ and $\underline{C}^I, \overline{C}^I$ closely approximate A_d^I.

The result of such learning can be improved by a test information system.

Our method for this problem, as described below, is based on bisimulation:

- Compute the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$, further denoted by $\{Y_{i_1},\ldots,Y_{i_k}\}$, together with concepts C_{i_t} characterizing Y_{i_t} (i.e. $C_{i_t}^I = Y_{i_t}$ for $1 \leq t \leq k$) as described in the previous subsection.
- Take $\underline{C} = C_{j_1} \sqcup \ldots \sqcup C_{j_h}$, where $j_1,\ldots,j_h$ are all the indices among $i_1,\ldots,i_k$ such that $Y_{j_t} \subseteq A_d^I$ for all $1 \leq t \leq h$.
- Take $\overline{C} = C_{j'_1} \sqcup \ldots \sqcup C_{j'_{h'}}$, where $j'_1,\ldots,j'_{h'}$ are all the indices among $i_1,\ldots,i_k$ such that $Y_{j'_t} \cap A_d^I \neq \emptyset$ for all $1 \leq t \leq h'$.
- Normalize $\underline{C}$ and $\overline{C}$, while preserving equivalence.

The pair $(\underline{C},\overline{C})$, obtained as above, is a pair of concepts in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ that approximates A_d on I most closely (in the sense that $\underline{C}^I \subseteq A_d^I \subseteq \overline{C}^I$ and the sets $A_d^I \setminus \underline{C}^I$ and $\overline{C}^I \setminus A_d^I$ are the smallest ones).

The accuracy on I does not imply accuracy on other information systems. Following the Ockham's razor principle, we pay attention to simplicity of $(\underline{C},\overline{C})$ in order to increase their overall accuracy. Here, we can use the following techniques:

- We use *LargestContainer* (see Subsection 19.5.3) to obtain a simpler form for $\underline{C}$.
- In the granulation process of Δ^I, we can stop as soon as the current partition is good enough according to some measure, and use it to compute $\underline{C}$ and $\overline{C}$.
- Using a test information system we can simplify $\underline{C}$ and $\overline{C}$ (without preserving equivalence) by applying different kinds of simplification as discussed in the previous subsection, taking into account the accuracies of the lower and upper approximations on the test information system and the relation between them.

Example 19.9. Consider again the information system given in Example 19.5. We want to learn a concept definition or a concept approximation for the set $X = \{Alice, Bob, Claudia\}$ in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger = \{Adult, has\_child\}$ and $\Phi^\dagger = \emptyset$. This task can be realized as follows:

1. $Y_1 := \Delta^I$, $partition := \{Y_1\}$
2. partitioning Y_1 by *Adult*:
   - $Y_2 := \{Alice, Bob, Claudia, Dave, George\}$, $C_2 := Adult$
   - $Y_3 := \{Eva, Frank\}$, $C_3 := \neg Adult$
   - $partition := \{Y_2, Y_3\}$
3. partitioning Y_2 by $\exists has\_child.\top$:
   - $Y_4 := \{Alice, Bob, Claudia, Dave\}$, $C_4 := C_2 \sqcap \exists has\_child.\top$
   - $Y_5 := \{George\}$, $C_5 := C_2 \sqcap \neg\exists has\_child.\top$
   - $partition := \{Y_3, Y_4, Y_5\}$
4. partitioning Y_4 by $\exists has\_child.C_2$ (we use the selector $\exists has\_child.C_2$ instead of $\exists has\_child.C_4$ because it is simpler and has the same effect):
   - $Y_6 := \{Alice, Bob\}$, $C_6 := C_4 \sqcap \exists has\_child.C_2$
   - $Y_7 := \{Claudia, Dave\}$, $C_7 := C_4 \sqcap \neg\exists has\_child.C_2$
   - $partition := \{Y_3, Y_5, Y_6, Y_7\}$
5. The current partition cannot be granulated anymore. (It corresponds to $\sim_{\Sigma^\dagger, \Phi^\dagger, I}$.)
6. Since only Y_6 from the current partition $\{Y_3, Y_5, Y_6, Y_7\}$ is a subset of X, the lower approximation of X is characterized by $C_6 = Adult \sqcap \exists has\_child.\top \sqcap \exists has\_child.Adult$, which can be simplified to $Adult \sqcap \exists has\_child.Adult$.
7. Since only Y_6 and Y_7 from the current partition $\{Y_3, Y_5, Y_6, Y_7\}$ overlap with X, the upper approximation of X is characterized by $C_6 \sqcup C_7$, which can be simplified to $C_4 = Adult \sqcap \exists has\_child.\top$. □

19.6 Conclusions

In the current chapter, we have studied roughification methods allowing one to construct indiscernibility relations on the basis of background knowledge. We have first studied indiscernibility based on similarity relations, showing that such relations can be turned into equivalence relations providing more accurate approximations. Next, we introduced roughifications based on relational databases and finally terminological roughifications, where indiscernibility coincides with indiscernibility by formulas of considered description logics. To our best knowledge, the proposed techniques and their applications are novel. It is worth emphasizing that our work is a pioneering one that uses bisimulation for machine learning in the context of description logics.

We have considered applications of the proposed techniques for improving accuracy of approximations, granulating relational databases as well as in concept learning and concept approximations. The last mentioned application areas have usually been studied in the context of information systems using only attributes (and sometimes also "external" relational structures) [21, 20]. In approaches based on RS information systems, concepts are usually characterized by formulas built from

unary predicates (corresponding to attributes), using propositional connectives. On the other hand, concept learning and concept approximation in information systems based on description logics require new methods and algorithms. Most ideas for them may be inspired from the traditional ones (like the ones based on decision rules, decision trees, reducts, and local reducts). However, additional ideas are needed to generalize such approaches to the case of description logics. We have shown that bisimulation is a good starting point.

As interesting continuations of the research reported in this chapter we consider extensions of roughifications techniques by considering other logical formalisms.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1996)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): Description Logic Handbook. Cambridge University Press (2002)
3. Baader, F., Nutt, W.: Basic description logics. In: Baader et al. [2], pp. 47–100
4. Borgida, A., Lenzerini, M., Rosati, R.: Description logics for databases. In: Baader et al. [2], pp. 472–494
5. Divroodi, A., Nguyen, L.: On bisimulations for description logics. CoRR abs/1104.1964 (2011) (appeared also in the proceedings of CS&P 2011, pp. 99–110)
6. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge representation techniques. A rough set approach. STUDFUZZ, vol. 202, Springer (2006)
7. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. In: International Joint Conference on AI, IJCAI 2001, pp. 145–151 (2000)
8. Doherty, P., Łukaszewicz, W., Szałas, A.: Tolerance spaces and approximative representational structures. In: Proceedings of 26th German Conference on Artificial Intelligence. Springer (2003)
9. Doherty, P., Szałas, A.: On the Correspondence between Approximations and Similarity. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 143–152. Springer, Heidelberg (2004)
10. Doherty, P., Szałas, A.: A correspondence framework between three-valued logics and similarity-based approximate reasoning. Fundamenta Informaticae 75(1-4) (2007)
11. Fanizzi, N., d'Amato, C., Esposito, F., Lukasiewicz, T.: Representing uncertain concepts in rough description logics via contextual indiscernibility relations. In: Proceedings of URSW 2008. CEUR Workshop Proceedings, vol. 423 (2008)
12. Greco, S., Matarazzo, B., Słowiński, R.: Fuzzy Similarity Relation as a Basis for Rough Approximations. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 283–289. Springer, Heidelberg (1998)
13. Hopcroft, J.: An $n \log n$ algorithm for minimizing states in a finite automaton (1971), ftp://reports.stanford.edu/pub/cstr/reports/cs/tr/71/190/CS-TR-71-190.pdf
14. Lin, F.: On strongest necessary and weakest sufficient conditions. In: Cohn, A., Giunchiglia, F., Selman, B. (eds.) Proc. 7th International Conf. on Principles of Knowledge Representation and Reasoning, KR 2000, pp. 167–175. Morgan Kaufmann Pub., Inc. (2000)

15. Lin, T.: Granular computing on binary relations I, II. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1: Methodology and Applications. STUDFUZZ, vol. 17, pp. 107–140. Physica-Verlag (1998)
16. Nardi, D., Brachman, R.J.: An introduction to description logics. In: Baader et al. [2], pp. 5–44
17. Nguyen, H., Skowron, A., Stepaniuk, J.: Granular computing: A rough set approach. Computational Intelligence 17, 514–544 (2001)
18. Nguyen, L.: An efficient tableau prover using global caching for the description logic $\mathcal{ALC}$. Fundamenta Informaticae 93(1-3), 273–288 (2009)
19. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about. Data. Kluwer Academic Publishers, Dordrecht (1991)
20. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Inf. Sci. 177(1), 41–73 (2007)
21. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
22. Peters, J., Ramanna, S., Skowron, A., Stepaniuk, J., Suraj, Z., Borkowski, M.: Sensor fusion: A rough granular approach. In: Proc. of the Joint 9th International Fuzzy Systems Association World Congress and 20th NAFIPS International Conference, pp. 1367–1371 (2001)
23. Polkowski, L., Skowron, A.: Towards adaptive calculus of granules. In: Zadeh, L., Kacprzyk, J. (eds.) Computing with Words in Information/Intelligent Systems, vol. 1-2, pp. 201–227. Physica-Verlag (1999)
24. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
25. Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular computing. International Journal of Intelligent Systems 16/1, 57–86 (2001)
26. Skowron, A., Stepaniuk, J.: Information granules and rough-neurocomputing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neuro Computing: Techniques for Computing with Words, pp. 43–84. Springer (2004)
27. Ślęzak, D.: Rough sets and few-objects-many-attributes problem: The case study of analysis of gene expression data sets. In: FBIT, pp. 437–442. IEEE Computer Society (2007)
28. Ślęzak, D., Wróblewski, J.: Roughfication of Numeric Decision Tables: The Case Study of Gene Expression Data. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 316–323. Springer, Heidelberg (2007)
29. Słowiński, R., Vanderpooten, D.: Similarity relation as a basis for rough approximations. In: Wang, P. (ed.) Advances in Machine Intelligence & Soft Computing, pp. 17–33. Bookwrights, Raleigh (1997)
30. Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. on Data and Knowledge Engineering 12(2), 331–336 (2000)
31. Szałas, A.: Second-order reasoning in description logics. Journal of Applied Non-Classical Logics 16(3-4), 517–530 (2006)

Chapter 20
How Near Are Zdzisław Pawlak's Paintings?
Study of Merotopic Distances between Digital Picture Regions-of-Interest

James F. Peters

How Near?
How near to the bark of a tree are the drifting snowflakes,
swirling gently round, down from winter skies?
...

—Z. Pawlak and J.F. Peters, Winter 2002 [22].

Abstract. This chapter commemorates the work of Zdzisław Pawlak as a painter with the focus on the subtleties that come to light in considering the symmetries in his paintings. Specifically, this chapter considers how merotopic distance functions can be used as an aid to visual perception in determining the nearness of Zdzisław Pawlak's paintings. Eventually, the study of the resemblance of perceptual fragments found in nature (*e.g.*, collections of falling snow flakes) in the poem *How Near?* by Z. Pawlak and J.F. Peters in 2002 led to the discovery of descriptively near sets by J.F. Peters in 2007 and a merotopological approach to measuring the nearness of collections of subsets recently introduced by J.F. Peters, S.A. Naimpally and S. Tiwari. The main contribution of this chapter is the introduction of an approach to measuring the nearness or apartness of Z. Pawlak's paintings in terms of the merotopic distances between collections of neighbourhoods in digital picture regions-of-interest. This study includes a consideration of ε-approach nearness spaces as frameworks in the search for patterns in digital pictures. An application of the proposed approach to measuring visual image nearness is reported relative to resemblances between Z. Pawlak's paintings of waterscapes that span more than a half century, starting in 1954. This study offers a partial answer to the question *How near are Zdzisław Pawlak's paintings?*

James F. Peters
Computational Intelligence Laboratory
University of Manitoba,
Department of Electrical & Computer Engineering,
75A Chancellor's Circle, EITC Room E1-526
Winnipeg, Manitoba, R3T 3E2, Canada
e-mail: jfpeters@ee.umanitoba.ca

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 545–568.
springerlink.com

Keywords: Approach nearness, description, distance, merotopy, painting, perceptual image analysis, visual neighbourhood.

20.1 Introduction

Zdzisław Pawlak was not only a pioneer in computer engineering, applied mathematics, machine learning, theoretical computer science, and philosophy, he was also a painter in his own right. His paintings capture various moments in the four seasons in the Polish countryside and offer his perception of the symmetries and singular beauty of the woods, gently rolling terrain, farmland furrows, waterways, lakes, and shorelines found in parts of Poland. He was a gifted painter. In fact, he was extraordinarily perceptive in his rendition of the changing face of the woods that border lakes and waterways that he visited.

His paintings reflect his fascination with seasonal changing colours of woodlands and the subtle shapes of watershadows thrown across the water by the sunlight on trees overhanging lake shorelines as well as the shadows made by marshland reeds and cattails shooting out of waterway borders. His interest in and perceptions of the four seasons in his paintings that span more than half a century, starting during the early 1950s, parallel his discovery of the secrets of rough sets during the late 1970s and early 1980s. In some sense, the manner in which Zdzisław Pawlak painted border regions of elongated watershadows thrown across the water by tall marshland reeds, shrubs and trees is reminiscent of the basic idea of set approximations in rough set theory. In effect, a consideration of Pawlak's perceptions of the compositions, symmetries and irregularities in border regions in a typical countryside as well as his portrayal of sunlight on the trees in the formation of watershadows and the reflections of light in lakeland shoreline scenes tend to reveal the plasticity and richness not only of nature itself but also the depth, breadth, and richness of rough sets themselves.

This chapter considers how to measure the nearness of neighbourhoods of points as well as the nearness of regions-of-interest (ROIs) in paintings by Zdzisław Pawlak. Briefly, each ROI is a collection of neighbourhoods of ROI points. Each ROI neighbourhood N_x considered in this study is a neighbourhood of a particular ROI point x (called a *reference point*) so that N_x contains points that are visually similar to the reference point appearance.

Measuring the perceptual nearness of visual images has recently led to a new form of approach distance between pairs of nonempty sets in $\mathcal{P}(X)$ [30, 34] (a variation of the original distance between points in X and nonempty sets in $\mathcal{P}(X)$ introduced by R. Lowen in 1997 [14]) and merotopies inspired by M. Katětov's work on topologizing parts of sets [9]. Work on a basis for near sets began in 2002, motivated by image analysis and inspired by a study of the perception of the nearness of physical objects carried out in cooperation with Zdzisław Pawlak in [22]. This initial work on the perception of typical scenes in nature, especially winter scenes, led to the introduction of near sets [27], elaborated in [26, 36, 6, 7, 43, 42, 28, 29, 37],

inspired by pioneering work on proximity spaces by S.A. Naimpally [16, 17, 19, 18] and my collaboration with A. Skowron and J. Stepaniuk on the nearness of objects and information granulation [31, 32, 33, 38].

This chapter considers the of Z. Pawlak's paintings in three different ways. First, a form of the indiscernibility relation is used to segment sample images. This simple approach to comparing paintings yields surprisingly remarkable results. Just from the segmentations alone, it is possible to see some fundamental painting techniques that Pawlak used in his paintings. In addition, it is possible to detect patterns in the segmentations that suggest the nearness of the autumn and springtime paintings of the waterscapes that are considered.

Second, visual neighbourhoods of points in Pawlak's paintings are considered. In doing this, the foundation for a study of what are known as ε-approach merotopic distances between collections of sets arises. In addition, a distinction is made between traditional spherical neighbourhoods (also called open balls) and the more recent visual neighbourhoods of points. In Fig. 20.1, the members of a visual neighbourhood $N_{\mathbf{y}}$ of a point y are sufficiently near (visually) in appearance to y (represented by a feature vector $\mathbf{y}$) to qualify for membership in the neighbourhood. Sample visual neighbourhoods can be seen in autumn waterscape by Pawlak in Fig. 20.3, especially in Fig. 20.3.3. Perceptually, the points in a visual neighbourhood are indistinguishable in appearance. This gives rise to the notion of the *distinctness of visual points* in separate visual neighbourhoods.

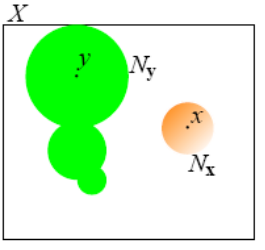


Fig. 20.1. Sample $N_{\mathbf{x}}, N_{\mathbf{y}}$

Let $\varepsilon \in (0, \infty]$. A *description of a point* x is a feature vector $\mathbf{x}$ containing numbers representing feature values extracted from a visual object such as a picture element (pixel) in a painting. Further, let $\mathbf{x}, \mathbf{y}$ denote the descriptions of points $x, y \in X$, respectively. A point x is visually distinct from a point y if, and only if, $d(\mathbf{x}, \mathbf{y}) > \varepsilon$, where d denotes a distance function. In effect, visually distinct points belong to non-intersecting visual neigbourhoods. Let $N_{\mathbf{x}} and N_{\mathbf{y}}$ denote a pair of visual neighbourhoods of points x, y, respectively (see Fig. 20.1, *for example*, for a sample representation of such neighbourhoods). In Fig. 20.1, $N_{\mathbf{x}}$ has the conventional shape of a spherical neighbourhood but $N_{\mathbf{y}}$ has the more usual shape of a visual neighbourhood, where the points of $N_{\mathbf{y}}$ are arranged asymmetrically in relation to the point y. Such visual neighbourhoods are commonly found in Pawlak's rendition of such things as watershadows found in many of his paintings of Polish woodland scenes.

The nearness of visual neighbourhoods can be measured with a Čech distance $D(A, B)$ between sets A and B (in the context of Pawlak's paintings, the Čech distance between visual neighbourhoods is measured to quantify the nearness of painting regions represented by selected visual neighbourhoods). A Čech distance between sets is the greatest lower bound of the distances between pairs of set members [41, §18.A.2]. This approach to measuring distances between sets in Pawlak's paintings works well because it captures minimal distances between set objects.

This approach also *implicitly* ushers in a new view of the nearness of rough sets, since many subsets in Pawlak's paintings are, in fact, visual rough sets.

Finally, the third approach to measuring the nearness of Pawlak's paintings is given in terms of approach merotopic distance and ε-approach nearness spaces. This leads to a more global view of the nearness of Pawlak's paintings, since merotopic distance functions and, in particular, ε-approach nearness, make it possible to measure the distance between collections of neighbourhoods. To give this brief study of Pawlak's paintings more focus, regions-of-interest in Pawlak's paintings are considered. A region of interest (ROI) is defined to be a collection of visual neighbourhoods. Then the nearness of Pawlak's paintings is quantified in terms of the ε-approach merotopic distance between ROIs.

20.2 Preliminaries

Let X be a nonempty ordinary set. The power set of X is denoted by $\mathcal{P}(X)$ and the set of all collections of subsets of $\mathcal{P}(X)$ is denoted by $\mathcal{P}^2(X)$. We write ABSA to denote the cardinality of A, where $A \subseteq X$. For collections $\mathcal{A}, \mathcal{B} \in \mathcal{P}^2(X)$, we say $\mathcal{A} \vee \mathcal{B} \doteqdot \{A \cup B : A \in \mathcal{A}, B \in \mathcal{B}\}$; $\mathcal{A}$ *corefines* $\mathcal{B}$ (written as $\mathcal{A} \prec \mathcal{B}$) if, and only if, for all $A \in \mathcal{A}$, there exists $B \in \mathcal{B}$ such that $B \subseteq A$. For $\alpha \in (0,\infty]$, the boundary set $A^{(\alpha)}$ is defined to be

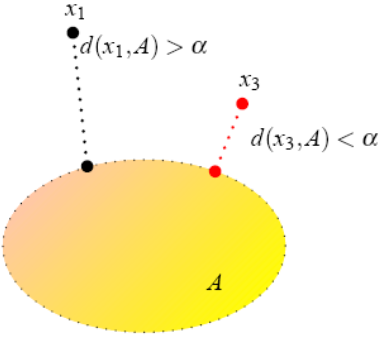


Fig. 20.2. Sample $A^{(\alpha)}$

$$A^{(\alpha)} \doteqdot \{x \in X : \delta(x,A) \leq \alpha\}.$$

Definition 20.1. A function $\delta : X \times \mathcal{P}(X) \longrightarrow [0,\infty]$ is called a *distance* on X [14, 15] if, for any $A, B \subseteq X$ and $x \in X$, the following conditions are satisfied.

(**D**.1) $\delta(x,\{x\}) = 0$,
(**D**.2) $\delta(x,\emptyset) = \infty$,
(**D**.3) $\delta(x,A \cup B) = \min\{\delta(x,A),\delta(x,B)\}$,
(**D**.4) $\delta(x,A) \leq \delta(x,A^{(\alpha)}) + \alpha$.

The pair (X,δ) is called an *approach space*.

Example 20.1. Let X be a nonempty set and let $x \in X, A \in \mathcal{P}(X)$. The distance function $d : X \times \mathcal{P}(X) \to [0,\infty]$ is defined by

$$d(x,A) = \inf\{\sigma(x,y) : y \in X\},$$

where $\sigma(x,y)$ is the standard distance between x and y. Distance $d(x,A)$ is the *lower distance* (*i.e.* greatest lower bound of the distances between the point x and the

points y in B) from F. Hausdorff [4] (see [5, §22, p. 128]). Observe that (X,d) is an approach space.

An approach space extends the usual notion of a point-based metric space to a space defined with a point-to-set-based distance function δ. In this space, the triangle inequality axiom $d(x,y) \leq d(x,z)+d(z,y)$ for all x,y in a metric space (X,d) is replaced by **D**.4 using the boundary set $A^{(\alpha)}$. A sample representation of a boundary set

$$A^{(\alpha)} = \{x_3\} \cup A.$$

is shown in Fig. 20.2. The set A is a member of $A^{(\alpha)}$, since $d(a,A) = 0$ for each $a \in A$ and, by definition, $0 < \alpha$.

Pawlak Autumn (3.2) Greyscale view (3.3) Segmentation

Fig. 20.3. Sample Pawlak Autum Painting Segmentation

Definition 20.2. A *generalized approach space* (X,ρ) [34, 35] is a nonempty set X equipped with a distance function $\rho : \mathcal{P}(X) \times \mathcal{P}(X) \longrightarrow [0,\infty]$ if, and only if, for all nonempty subsets $A,B,C \in \mathcal{P}(X)$, ρ satisfies properties (A.1)-(A.5), *that is*,

(**A**.1) $\rho(A,A) = 0$,
(**A**.2) $\rho(A,\emptyset) = \infty$,
(**A**.3) $\rho(A,B \cup C) = \min\{\rho(A,B),\rho(A,C)\}$,
(**A**.4) $\rho(A,B) = \rho(B,A)$,
(**A**.5) $\rho(A,B) \leq \rho(A,B^{(\alpha)}) + \alpha$, for every $\alpha \in [0,\infty]$,
where $B^{(\alpha)} \doteqdot \{x \in X : \rho(\{x\},B) \leq \alpha\}$.

It has been observed that the notion of distance in an approach space is closely related to the notion of nearness [11, 10, 40].

Definition 20.3. Čech Gap Distance
For a nonempty subset $A \in \mathcal{P}(X)$ and a nonempty set $B \in \mathcal{P}(X)$, define a *gap function* $D_\rho(A,B)$, a variation of the distance function introduced by E.Čech in his 1936–1939 seminar on topology [41] (see, also, [1, 4, 12]), where $D : \mathcal{P}(X) \times \mathcal{P}(X) \longrightarrow [0,\infty]$ is defined by

$$D(A,B) \doteqdot \begin{cases} \inf\limits_{a \in A} \{d(a,B)\}, & \text{if } A \text{ and } B \text{ are not empty,} \\ \infty, & \text{if } A \text{ or } B \text{ is empty.} \end{cases}$$

Observe that (X, D) is a generalized approach space.

20.3 Perceptual Indiscernibility Relation in Segmenting Paintings

Remarkably, an intuition about the nearness of Pawlak's paintings can be gleaned from various segmentations of a digital image obtained by using a description-based form of the original indiscernibility relation [21, 25, 24, 23] that is termed a *perceptual* indiscernibility relation $\sim_B$. Let

$$X = \{\text{points in a digital image}\},$$
$$\phi : X \to [0, \infty] \text{ probe function},$$
$$B = \{\phi : \phi(x) = \text{ feature value}\}.$$

Fig. 20.4. Sample Autumn Class

With this in mind, a relation $\sim_B$ is defined by

$$\sim_B = \{(x, y) \in X \times X : \forall \phi \in B, \phi(x) = \phi(y)\}$$

Let $x_{/\sim_B}$ denote an *equivalence class* containing x in a partition of a nonempty set X determined by the relation $\sim_B$. Then, an element $y \in X$ is a member of $x_{/\sim_B}$ if, and only if, the descriptions of x and y are the same, *that is*,

$$y \in x_{/\sim_B}, \text{ provided } \forall \phi \in B, \overbrace{\phi(x) = \phi(y)}^{\text{same descriptions}}.$$

A sample class extracted from the segmentation of the Autumn painting by Pawlak in Fig. 20.3.3 is shown in Fig. 20.4. The relation $\sim_B$ is inspired by the original work by Z. Pawlak on classifying objects by means of attributes [20]. Rather than using attributes, the relation $\sim_B$ is defined in terms of probe functions that offer a convenient way to describe visual objects such as picture elements (pixels) in digital images. The relation $\sim_B$ is termed *perceptual*, since the probe functions in B, in some sense, mimic human perception of sensed features of objects such as colour, texture and shape.

Segmentation is a fundamental part of image analysis. A digital image is segmented by separating the image into regions, where the objects in each region have similar feature values. The simplest form of segmentation results from identifying greyscale image regions with similar intensities. This is the approach used in the examples in this section. This is accomplished rather easily and very beneficially by employing the perceptual indiscernibility relation to obtain a partition of image pixel intensities.

Example 20.2. **How to Segment an Image with the NEAR system**
Using the NEAR system Eq option[1], import the Autumn painting shown in Fig. 20.3.1, choose subimage size $p = 3$, and choose one or more image features used in obtain an image segmentation and display of image classes.

For simplicity in this example, only the average greyscale feature (for $p \times p$ subimages) is chosen. The greyscale image for Pawlak's Autumn painting is shown in Fig. 20.3.2. A sample segmentation of Pawlak's painting is given in Fig. 20.3.3. In Fig. 20.3.3, each colour represents the points in an equivalence class (see, *e.g.*, the single class represented by the purple ■ splotches (splash of like-coloured pixels) in Fig. 20.4).

Using the NEAR system picture point (pixel) touch feature, use the cursor to touch a particular pixel of interest in an image showing a Pawlak painting. This pixel-selection operation results in the display of an image class, *that is*, the set of all 3×3 subimages with similar average pixel intensities. For a sample segmentation class, see Fig. 20.4. This particular class reveals Pawlak's extensive use of the same shading (pixel intensities) for trees and water. ■

Pawlak Springtime (5.2) Greyscale view (5.3) Segmentation

Fig. 20.5. Sample Pawlak Springtime Painting Segmentation

Example 20.3. **Sample Pawlak Springtime Painting Segmentation**
Using the NEAR system, import the Pawlak Springtime painting shown in Fig. 20.5.1, choose subimage size $p = 3$, and again, for simplicity, choose a single feature, namely average greyscale feature. The greyscale image for Pawlak's painting is shown in Fig. 20.5.2. The segmentation of Pawlak's painting is given in Fig. 20.5.3. Using the NEAR system display, use the cursor to touch a particular pixel of interest in the displayed image. This pixel-selection operation results in the

[1] 

NEAR system tabs

The NEAR system makes it possible to analyse digital images. For instance, image can be studied either singly using the Eq tab option to determine equivalence classes in segmentations or in pairs using the NI tab. To see the equivalence classes in an image, choose the Eq tab and use the load button to import an image into the system. The complete NEAR system, version 1.2, is available at http://wren.ece.umanitoba.ca

display of an image class, *that is*, the set of all 3×3 subimages with similar average pixel intensities. For a sample segmentation class, see Fig. 20.6. This particular class reveals Pawlak's extensive use of the same shading (pixel intensities) for trees, water as well as shoreline reeds shown in the foreground in Fig. 20.5.1. It can also be observed that higher intensities of light tend to dominate Pawlak's Springtime painting (see, *e.g.* the dominance of the lighter regions shown Fig. 20.5.2.

The dominance of lighter springtime regions in Fig. 20.5.1 contrasts sharply with the more sombre-shaded regions in the autumn painting in Fig. 20.3.1. In both Fig. 20.4 (Autumn painting class) and Fig. 20.6 (Springtime painting class), notice that colour intensities for the shoreline trees chosen by Pawlak are subtly different and reflect the changing angle of the sun in forming tree shadows (*i.e.* various combinations of light and dark of tree leaves, not painted individually but portrayed as the eye would see them from a distance as varying splotches of greens and blues). ■

Fig. 20.6. Sample Springtime Class

From the sample segmentations, one can begin to see shading patterns in Pawlak's paintings. For example, very intensities of light are used by Pawlak in the shading of parts of the distant shoreline trees separated by white light between the two groups of trees in the background, tree watershadows, and the shading of the foreground reeds. These patterns become more obvious in the study of neighbourhoods of points given in Sect. 20.4 and study of regions of interest (ROIs) in Sect. 20.6. In addition, by considering the varying Čech (and other) distances between neighbourhoods in ROIs in pairs of paintings, it is possible to gain a deeper knowledge of the subtleties in Pawlak's paintings, his use of shading to portray in a vivid way signs of Autumn and Spring.

20.4 Neighbourhoods in Paintings by Z. Pawlak

Neighbourhoods of points in Z. Pawlak's paintings are briefly considered in this section. Let A be a nonempty subset of topological space X. Recall that a neighbourhood of A is any subset of X that is an open set containing A [2, §1.2, p. 19]. A set A is *open* if, and only if, for each $x \in A$, all points *sufficiently near* x belong to A. In this study, a neighbourhood of a point x is an open set A such that all of the points in A that are sufficiently near x.

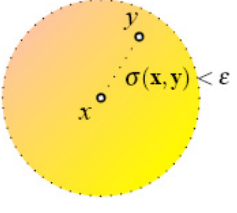


Fig. 20.7. Nbd N_x

Traditionally, nearness of points is measured in terms of the location of the points. Let $\sigma : X \times X :\rightarrow [0,\infty]$ denote the standard distance between points in X[2]. For $\varepsilon \in (0,\infty]$, a neighbourhood of $x \in X$ (denoted N_x) is the set of all $y \in X$ such that $\sigma(x,y) < \varepsilon$ (see, *e.g.* Fig. 20.7, where the distance $\sigma(x,y)$ between each pair x,y is less than ε in the neighbourhood). In that case, a neighbourhood is called an open ball [3, §4.1] or spherical neighbourhood [8, §1-4]. In the plane, the points in a spherical neighbourhood are contained in the interior of a circle.

In this study of Z. Pawlak's paintings, the focus is on what is known as a visual neighbourhood $N_{\mathbf{x}}$ of a point x. In such a neighbourhood, points have similar visual appearance, each represented by a feature vector $\mathbf{x}$ (*e.g.* a vector of feature values for colour, texture, shape) extracted from a particular picture point or picture element (pixel). This visual information is extracted from each pixel in the form of probe function values.

A visual image is viewed as a set of points, where each point is an image pixel. A nbd $N_{\mathbf{x}}$ of a point x is an open set A such that the visual information values extracted from all of the points in A are sufficiently near the appearance $\mathbf{x}$ of x. In its simplest form, a neighbourhood of a point x (ignoring point descriptions) is denoted simply by N_x (*cf.* W.J. Thron [39, ch.4]) and by $N_{\mathbf{x}}$ for a visual neighbourhood with reference point x, where the distance between neighbourhood points is less than a threshold $\varepsilon \in (0,\infty]$[3].

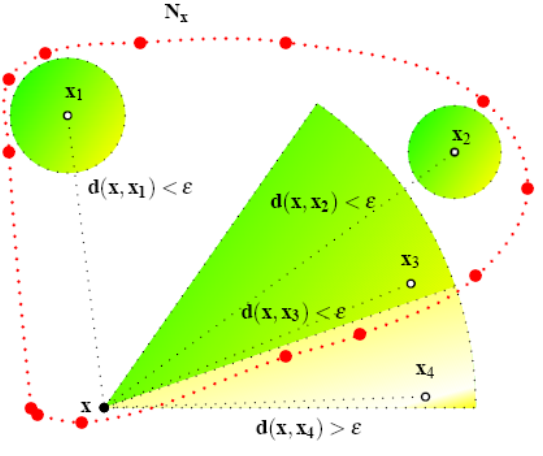


Fig. 20.8. Sample Visual Neighbourhood $\mathbf{N_x}$ Inside the ···●·····●·· Border

[2] *That is*, for $x,y \in X \subset \mathbb{R}, \sigma(x,y) = \mid x-y \mid$.

[3] The term *threshold* is preferred here, since ε is not a radius but rather an upper bound on the distance between each pair of pixel descriptions for pixels that belong to a visual neighbourhood. When only one probe function is used in pixel descriptions, then a visual neighbourhood is denoted by $N_{\phi(x)}$. Similarly, the term *reference point* is used instead of *centre* to call attention to the fact that a reference point is often not in the centre of a visual neighbourhood.

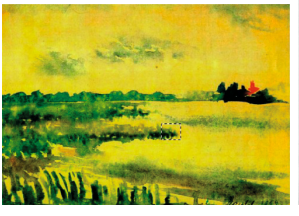
Let X denote a set of pixels in a digital image. Each pixel description is determined by a feature vector containing probe function ϕ values representing pixel visual information in a digital image. Let $y \in X$. For simplicity, a feature vector $\mathbf{y}$ for a pixel y (*i.e.* description of y) is defined by

$$\mathbf{y} = (\phi_1(y), \ldots, \phi_i(y), \ldots, \phi_n(y)), \text{ pixel description,}$$

where $\phi_i(y)$ is a feature value extracted from pixel y and n is the length of a pixel description. The distance between the description $\mathbf{y}$ of each pixel y and the description $\mathbf{x}$ of reference point x is less than a threshold ε in the neighbourhood $N_{\mathbf{x}}$. The details for the sample visual neighbourhood in Fig. 20.8 are given next.

$$\begin{aligned}
X &= \{\text{digital image pixels}\},\\
\phi &: X \to [0,\infty], \text{ (probe)},\\
\mathbf{x},\mathbf{y} &= \text{ descriptions of pixels } x, y \in X,\\
d(\mathbf{x},\mathbf{y}) &= \sum_{i=1}^{n} \text{ABS}\phi_i(x) - \phi_i(y), \text{ (Manhattan distance between pixel descriptions)},\\
x &\in X, \text{ (visual neighbourhood reference point)},\\
\varepsilon &\in (0,\infty], \text{ (sufficient nearness bound)},\\
N_{\mathbf{x}} &= \{y \in X : d(\mathbf{x},\mathbf{y}) < \varepsilon\} \text{ (perceptual neighbourhood)}.
\end{aligned}$$

At this point, we again can observe that the appearance of a visual neighbourhood of a reference point can be quite different from the appearance of a spherical neighbourhood of a point.



Pawlak Springtime

(9.2) Greyscale view

(9.3) Neighbourhood

Fig. 20.9. Pawlak 1954 Waterscape Painting

Example 20.4. **Sample Visual Nbd of a Point**
This example focuses on the visual neighbourhood $N_{\mathbf{x}}$ inside the $\cdots\bullet\cdots\bullet\cdots$ border in Fig. 20.8. Observe that the members in a visual neighbourhood of point have descriptions that are *sufficiently near* the visual neighbourhood centre. Let $\phi : X \to [0,\infty]$ be defined by

$$\phi(x) = \text{pixel intensity}.$$

For example, each of the points in the green-shaded regions in Fig. 20.8 have intensities that are very close to the intensity of the point x. By contrast, many points in the yellow-shaded region have higher intensities (*i.e.* more light) than the pixel at x, For example, consider the intensities of the points in the visual nbd represented by the green wedge-shaped region and some outlying green circular regions and the point x_4 in the yellow-shaded region in Fig. 20.8, where

$$\begin{aligned}
&\varepsilon = 5 \text{ (low-intensity threshold)},\\
&d(\mathbf{x},\mathbf{x}_1) < \varepsilon,\\
&d(\mathbf{x},\mathbf{x}_2) < \varepsilon,\\
&d(\mathbf{x},\mathbf{x}_3) < \varepsilon, \text{ but}\\
&d(\mathbf{x},\mathbf{x}_4) > \varepsilon, \text{ where } \phi(x_4) = \text{high intensity (close to pure white)}
\end{aligned}$$

In the case of the point x_4 in Fig. 20.8, the intensity is high (close to white), *that is*, $\phi(x_4) \sim 255$. By contrast the point x has low intensity (less light), *for example*, $\phi(x) \sim 100$. Assume $\varepsilon = 5$. Hence, $\text{ABS}\phi(x) - \phi(x_4) > \varepsilon$. As in the case of Z. Pawlak's paintings, the distance between feature vectors representing visual information extracted from image pixels can be sufficiently near (perceptually) the appearance of a reference point x, but the locations of the pixels themselves can be far apart, *that is*, not sufficiently near or *far apart*, if one considers the locations of the pixels. ■

Fig. 20.10. Sample $\mathbf{N}_{\mathbf{x}_0}$, ε=5

The pronounced visual similarity between intensities of the points of light used in filling in skylines and water surfaces can be frequently observed in different regions in Z. Pawlak's waterscapes. This is especially true in the earliest known painting by Pawlak, a waterscape showing a marshland in Poland that was probably done in the fall of the year (shown in Fig. 20.10). Before we look at the details and a sample visual neighbourhood in this remarkable painting, observe one splotch of red indicates a farm building along the distant horizon in the upper right-hand side of the painting. This tiny bit of red (see ■ splotch in Fig. 20.9.1) contrasts sharply with the dominant yellows, bits of orange, dark greens of the marshland reeds, grasses and shrubs, and the reflection of the late afternoon sun on water shown in yellow as an artist would see the reflections of the light on the water.

Example 20.5. **Sample Visual Neighbourhood in Pawlak's 1954 Waterscape**
This example calls attention to one of the visual neighbourhoods in Pawlak's 1954 waterscape (see Fig. 20.9.1). This is a watercolour that has faded over time. This 1954 waterscape is rendered as a greyscale image in Fig. 20.9.2. Let $\varepsilon = 5$ and obtain the neighbourhood shown in Fig. 20.9.3 and Fig. 20.10.

A single visual neighbourhood shown in Fig. 20.10. In this visual neighbourhood, $\mathbf{N}_{\mathbf{x_0}}$ shown in Fig. 20.10, the intensities of all points are sufficiently near the intensity of point x_0. The set of intensities in $\mathbf{N}_{\mathbf{x_0}}$ are displayed with splotches of pastel green ■ in Fig. 20.10, providing a nice example of an open set. Notice that the pixel intensities for large regions of the water and sky are quite similar. This is the case with the sample pixels (points of light) x_0, x_1, x_2, where the in $\text{ABS}\phi(x_0) - \phi(x_1) < \varepsilon$ and $\text{ABS}\phi(x_0) - \phi(x_2) < \varepsilon$. ■

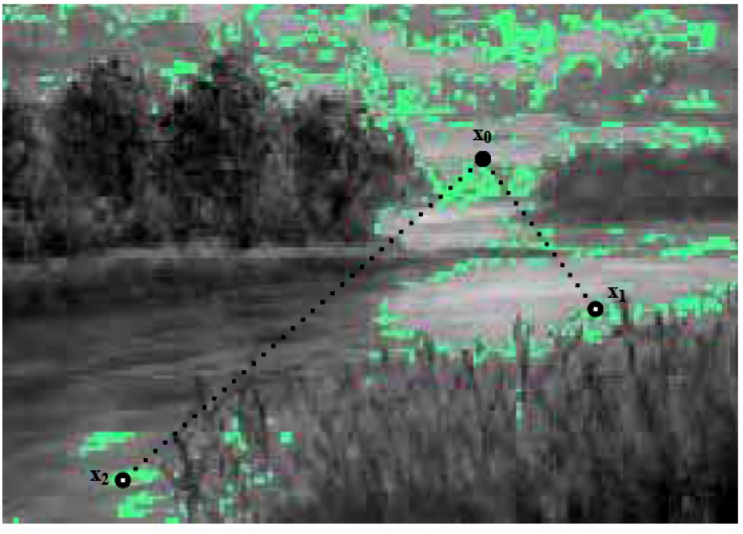
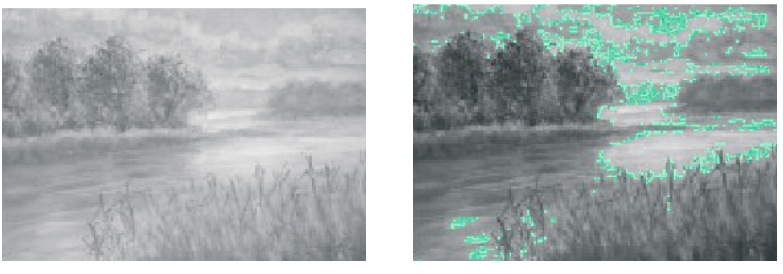


Fig. 20.11. Pawlak Visual Neighbourhood $\mathbf{N}_{\mathbf{x_0}}$, ε=5



Greyscale view (12.2) Sample Visual Nbd

Fig. 20.12. Greyscale Views of Springtime Painting

The pronounced visual similarity between intensities of the points of light in Pawlak's paintings can be found in different regions more recent Pawlak waterscapes. Consider, next, a recent (sometime before 2006) springtime waterscape by Pawlak shown as a greyscale image in Fig. 20.12.1. Before we consider one of the remarkable visual neighbourhoods in this painting, recall that a particular visual neighbourhood can be derived by using the cursor touch feature available in the NEAR system.

Example 20.6. **Visual Nbd in a Recent Waterscape by Z. Pawlak**
Let $\varepsilon = 5$ and obtain the single visual neighbourhood shown in Fig. 20.11. Again, notice that the pixel intensities a number of different regions of the water and sky are quite similar. This is the case with the sample pixels (points of light) x_0, x_1, x_2 shown in Fig. 20.11, where the in $\text{ABS}\phi(x_0) - \phi(x_1) < \varepsilon$ and $\text{ABS}\phi(x_0) - \phi(x_2) < \varepsilon$. In other words, the intensities of all points in the visual nbd in Fig. 20.11 are sufficiently near the intensity $\phi(x_0)$. ■

20.5 ε-Approach Nearness

This section briefly introduces what is known as ε-approach nearness spaces. Such a space provides a framework that is useful in finding pairs of collections of subsets that are sufficiently near or apart. An application of this form of nearness study framework is given in the sequel in terms of a study of regions-of-interest in Pawlak's paintings. Let $\mathcal{A}, \mathcal{B} \in \mathcal{P}^2(X)$, then define

$$\mathcal{A} \vee \mathcal{B} \doteqdot \{A \cup B : A \in \mathcal{A}, B \in \mathcal{B}\},$$
$$\mathcal{A} \prec \mathcal{B} \Leftrightarrow \forall A \in \mathcal{A}, \exists B \in \mathcal{B} : B \subseteq A \textit{ that is, } \mathcal{A} \text{ corefines } \mathcal{B}.$$

Assume $\varepsilon \in (0, \infty]$. An ε-*approach merotopy* on X is a function $\nu : \mathcal{P}^2(X) \times \mathcal{P}^2(X) \to [0, \infty]$ provided for any collections $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathcal{P}^2(X)$, we have

(**AN**.1) $\mathcal{A} \prec \mathcal{B} \Longrightarrow \nu(\mathcal{C}, \mathcal{A}) \leq \nu(\mathcal{C}, \mathcal{B})$,
(**AN**.2) $(\bigcap \mathcal{A}) \cap (\bigcap \mathcal{B}) \neq \emptyset \Longrightarrow \nu(\mathcal{A}, \mathcal{B}) < \varepsilon$,
(**AN**.3) $\nu(\mathcal{A}, \mathcal{B}) = \nu(\mathcal{B}, \mathcal{A})$ and $\nu(\mathcal{A}, \mathcal{A}) = 0$,
(**AN**.4) $\emptyset \in \mathcal{A} \Longrightarrow \nu(\mathcal{C}, \mathcal{A}) = \infty$,
(**AN**.5) $\nu(\mathcal{C}, \mathcal{A} \vee \mathcal{B}) \geq \nu(\mathcal{C}, \mathcal{A}) \wedge \nu(\mathcal{C}, \mathcal{B})$.

The pair (X, ν) is called an ε-*approach merotopic space*. There is continuing interest in the topological closure of a nonempty set (see, *e.g.* [11, 13]). For an ε-approach merotopic space (X, ν), the function $\text{cl}_\nu : \mathcal{P}(X) \longrightarrow \mathcal{P}(X)$ defined by

$$cl_\nu(A) = \{x \in X : \nu(\{\{x\}\}, \{A\}) < \varepsilon\}, \text{ for all } A \subseteq X$$

satisfies the following properties for all $A, B \in \mathcal{P}(X)$:

(**cl**.1) $\text{cl}_\nu(\emptyset) = \emptyset$,

(**cl**.2) $A \subset \mathrm{cl}_{\nu}(A)$,
(**cl**.3) $\mathrm{cl}_{\nu}(A \cup B) = \mathrm{cl}_{\nu}(A) \cup \mathrm{cl}_{\nu}(B)$.

That is, cl_{ν} is a Čech topological closure of a set [41, Def. 15.A.1]. Let $cl_{\nu}(\mathcal{A}) = \{cl_{\nu}(A) : A \in \mathcal{A}\}$. An ε-approach merotopy ν on X is called an ε-*approach nearness* on X for $\mathcal{A}, \mathcal{B} \in \mathcal{P}^2(X)$, provided

(**AN**.6) $\nu(cl_{\nu}(\mathcal{A}), cl_{\nu}(\mathcal{B})) \geq \nu(\mathcal{A}, \mathcal{B})$.

Then cl_{ν} also satisfies

(**cl**.4) $cl_{\nu}(cl_{\nu}(A)) = cl_{\nu}(A)$.

That is, cl_{ν} is a Kuratowski closure operator on X [13, Def. 2.2].

Example 20.7. The function $\nu_D : \mathcal{P}^2(X) \times \mathcal{P}^2(X) \longrightarrow [0, \infty]$ defined as

$$\nu_D(\mathcal{A}, \mathcal{B}) \doteqdot \sup_{A \in \mathcal{A}, B \in \mathcal{B}} D(A, B), \text{ where } \nu_D(\mathcal{A}, \mathcal{A}) \doteqdot \sup_{A \in \mathcal{A}} D(A, A) = 0$$

satisfies (AN.1)-(AN.5). Hence, ν_D is an ε-approach merotopy on X. Let $cl_{\nu_D}(\mathcal{A}) = \{cl_{\nu_D}(A) : A \in \mathcal{A}\}$, where cl_{ν_D} is defined by

$$cl_{\nu_D}(A) = \{x \in X : \nu_D(\{\{x\}\}, \{A\}) < \varepsilon\}, \text{ for all } A \subseteq X.$$

Then ν_D satisfies (AN.6). Hence, (X, ν_D) is an ε-approach nearness space. ■

Remark 20.1. **ε-approach nearness**
It is now possible to consider what is known as ε-approach nearness. The utility of this form of nearness is considerable in image analysis. In this section, an application of ε-approach nearness is given in a study of Z. Pawlak's waterscapes. Let $\varepsilon \in (0, \infty]$. Then the function $\nu_d : \mathcal{P}^2(X) \times \mathcal{P}^2(X) \longrightarrow [0, \infty]$ defined as: for $\mathcal{A}, \mathcal{B} \in \mathcal{P}^2(X)$, $\nu_d(\mathcal{A}, \mathcal{B}) = 0$, if $(\bigcap \mathcal{A}) \cap (\bigcap \mathcal{B}) \neq \emptyset$, and $\nu_d(\mathcal{A}) = \infty$, otherwise, is an ε-approach nearness on X and $cl_{\nu_d}(A) = A$, for all $A \subseteq X$. We call (X, ν_d) a *discrete* ε-*approach nearness space*. Further, the function $\nu_i : \mathcal{P}^2(X) \times \mathcal{P}^2(X) \longrightarrow [0, \infty]$ defined as: for $\mathcal{A}, \mathcal{B} \in \mathcal{P}^2(X)$, $\nu_i(\mathcal{A}, \mathcal{B}) = 0$, if $\emptyset \notin \mathcal{A}$ or $\emptyset \notin \mathcal{B}$, and $\nu_i(\mathcal{A}) = \infty$, otherwise, is an ε-approach nearness on X and $cl_{\nu_i}(A) = X$, for all nonempty subsets A of X. We call (X, ν_i) an *indiscrete* ε-*approach nearness space*.

Definition 20.4. Let $\mathcal{C} \in \mathcal{P}^2(X)$ and let (X, ν) be an ε-approach nearness space. Then $\mathcal{C}$ is a ν-cluster, provided the following conditions are satisfied:
(i) $C, D \in \mathcal{C} \Longrightarrow \nu(\{C\}, \{D\}) < \varepsilon$,
(ii) $\nu(\{A\}, \{C\}) < \varepsilon$, for all $C \in \mathcal{C} \Longrightarrow A \in \mathcal{C}$,
(iii) $C \cup D \in \mathcal{C} \Longrightarrow C \in \mathcal{C}$ or $D \in \mathcal{C}$.

Example 20.8. [40]. Let (X, ν) be an ε-approach nearness space. Denote $\mathfrak{e}(x) = \{A \subseteq X : x \in cl_{\nu}(A)\}$, $x \in X$. Then $\mathfrak{e}(x)$ is a ν-cluster, for all $x \in X$. ■

Property 20.1. **ε-Approach Nearness Separation Axiom**. An ε-approach nearness space (X, ν) is called *separated*, provided $\nu(\{\{x\}, \{y\}\}) < \varepsilon \Longrightarrow x = y$.

Definition 20.5. An ε-approach nearness space (X,ν) is said to be *complete* if, and only if, $\bigcap cl_\nu(\mathcal{A}) \neq \emptyset$, for all ν-clusters $\mathcal{A} \in \mathcal{P}^2(X)$

Remark 20.2. In an extended metric space, Definition 20.5 reduces to the usual definition of completion, *that is*, a metric space is complete if, and only if, each Cauchy sequence is convergent. Let (X,ν) be an ε-approach nearness space, let X^* be the set of all ν-clusters and let $f : X \longrightarrow X^*$ be defined by $f(x) = \mathfrak{e}(x)$. For $\Omega, \Im \in \mathcal{P}^2(X^*)$, define the function $\nu^* : \mathcal{P}^2(X^*) \times \mathcal{P}^2(X^*) \longrightarrow [0,\infty]$ by

$$\nu^*(\Omega,\Im) = \nu(\bigcup\{\bigcap\omega : \omega \in \Omega\}, \bigcup\{\bigcap\tau : \tau \in \Im\}).$$

Theorem 20.1. (X^*,ν^*) *is a complete separated* ε*-approach nearness space.*

The proof of Theorem 20.1 is given in [35] and not repeated, here. There are many instances of ε-approach nearness on X just as there are many instances of ε-approach spaces [14] and metric spaces on X.

Let $\Im_1$ and $\Im_2$ be a pair of digital images and let X be the set of all pixels of $\Im_1$ and $\Im_1$, *that is*, $X = \Im_1 \cup \Im_2$. If we want to compare these two images, then we choose a collection $\mathcal{A}$ of subimages from $\Im_1$ and another collection $\mathcal{B}$ of subimages from $\Im_2$. Then $\mathcal{A},\mathcal{B} \in \mathcal{P}^2(X)$. Here feature values are considered in measuring the nearness of the images [34]. We say that $\mathcal{A}$ is similar (near) to $\mathcal{B}$ (written as $\nu(\mathcal{A},\mathcal{B}) = 0$) if and only if there exists subimages $A \in \mathcal{A}$ and $B \in \mathcal{B}$ such that $\nu(\{A\},\{B\}) = 0$, where ν is an approach merotopy on X (cf. [15, 34]). Practically, this is a rare situation. For example, if we consider the feature *colour* in comparing $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{A}$ and $\mathcal{B}$ can have many different shades of the same colour (say blue). So, this observation leads us to consider $\mathcal{A}$ and $\mathcal{B}$ being *sufficiently near* with respect to the feature 'color'; *that is*, $\nu(\mathcal{A},\mathcal{B}) < \varepsilon$, where $\varepsilon \in (0,\infty]$. Thus a function that measures sufficient nearness is required. A recognition of the presence of sufficient nearness in comparing collections of objects motivated the axiomatization of ε-approach merotopic spaces, where $\varepsilon \in (0,\infty]$. That is, an ε-approach merotopy measures the sufficient nearness of two objects. Images $\Im_1$ and $\Im_2$ are said to be sufficiently near if, and only if, there exist subcollections $\mathcal{A} \in \mathcal{P}^2(\Im_1)$ and $\mathcal{B} \in \mathcal{P}^2(\Im_2)$ such that $\nu(\mathcal{A},\mathcal{B}) < \varepsilon$.

Example 20.9. Let $\Im_1$ and $\Im_2$ be two digital images and $X = \Im_1 \cup \Im_2$. A nonempty collection $\mathcal{A} \in \mathcal{P}^2(\Im_1)$ is descriptively ε-near a nonempty collection $\mathcal{B} \in \mathcal{P}^2(\Im_2)$ if, and only if, the description-based merotopy

$$\nu_{D_{\rho_{\|\cdot\|}}}(\mathcal{A},\mathcal{B}) \doteq \sup_{A\in\mathcal{A},B\in\mathcal{B}} D_{\rho_{\|\cdot\|}}(A,B) < \varepsilon,$$

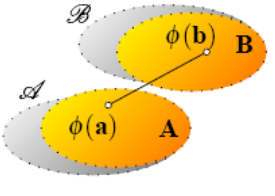


Fig. 20.13. ε-Near Images

for a chosen $\varepsilon \in (0,\infty]$. The distance function $\rho_{\|\cdot\|} : \mathbb{R}^n \times \mathbb{R}^n \to [0,\infty]$ is defined, by the $\|\cdot\|_1$ norm called the taxicab distance, *that is*, $\rho_{\|\cdot\|}(\mathbf{a},\mathbf{b}) = \| a - b \|_1 = \sum_{i=1}^{n} |a_i - b_i|$. For example, let $\mathbf{a},\mathbf{b}$ denote a pair of n-dimensional vectors of numbers that are

positive real values representing intensities of light reflected from objects in a visual field, *i.e.*, $\mathbf{a} = (a_1, \ldots, a_i, \ldots, a_n), \mathbf{b} = (b_1, \ldots, b_i, \ldots, b_n)$ such that $a_i, b_i \in \mathbb{R}^{0+}$. For this reason, we introduce a form of Čech gap functional $D_{\rho_{\|\cdot\|}}$ such that $D_{\rho_{\|\cdot\|}} : \mathcal{P}(X) \times \mathcal{P}(X) \to [0, \infty]$ is defined by

$$D_{\rho_{\|\cdot\|}}(A,B) \doteqdot \begin{cases} \inf\{\rho_{\|\cdot\|}(\mathbf{a},\mathbf{b}) : a \in A, b \in B\}, & \text{if } A \text{ and } B \text{ are not empty,} \\ \infty, & \text{if } A \text{ or } B \text{ is empty.} \end{cases}$$

Then, for example, consider

$$\nu_{D_{\rho_{\|\cdot\|}}}(\mathcal{A},\mathcal{B}) \doteqdot \begin{cases} \inf\left\{ \sup_{A\in\mathcal{A}, B\in\mathcal{B}} D_{\rho_{\|\cdot\|}}(A,B) \right\}, & \text{if } \mathcal{A} \text{ and } \mathcal{B} \text{ are not empty,} \\ \infty, & \text{if } \mathcal{A} \text{ or } \mathcal{B} \text{ is empty.} \end{cases}$$

The egg-shaped regions labelled A, B in the foreground represent descriptively similar neighbourhoods in digital images. If we consider only greylevel intensity of the pixels in A, B, then, by definition, for some choices of ε, $D_{\rho_{\|\cdot\|}}(A,B) < \varepsilon$. The pair of • bullets in Fig. 20.13 each has a feature value extracted with probe ϕ [27, 26]. Observe that $(X, \nu_{D_{\rho_{\|\cdot\|}}})$ is an ε-approach nearness space.

Table 20.1. Comparison Between Pawlak Waterscapes

feature(s)	Pawlak 1954 nbd vs. Recent spring nbd	ε-value	$D_{\rho_{\|\cdot\|}}$-value
Intensity	sky-water nbd (Fig. 20.10 vs. Fig. 20.11)	10	4.00
Intensity	sky-water nbd (Fig. 20.10 vs. Fig. 20.11)	5	0.00

It helps to experiment with measurements of the distance between a single pair of very simple collections of subsets, before considering ε-approach nearness measurements in largescale settings provided by Z. Pawlak's paintings. The following example suggests how to do this.

Example 20.10. **Sample Nearness Measurements Between Collections**
In this example, we first briefly consider measurement of the distance between single pairs of neighbourhoods in comparing digital pictures. Assume

$$A = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}, B = \begin{vmatrix} 5 & 6 \\ 7 & 8 \end{vmatrix}$$

represent a pair of tiny greyscale images A, B, each containing with four pixels with the indicated intensities. Also assume that A, B represent a pair of ROIs extracted with from a pair of larger greyscale images. Next, extract collections $\mathcal{A}, \mathcal{B}$ from the ROIs

$$\begin{aligned}
\mathcal{A} = \{A_1, A_2\}, \mathcal{B} = \{B_1, B_2, B_3\}, \text{ where} \\
A_1 = \{1,2\}, A_2 = \{3,4\}, \\
B_1 = \{5,6\}, B_2 = \{7,8\}, B_3 = \{9,10\} \\
D(B_1, A_1) = \inf\{\inf\{d(5,1), d(5,2)\}, \inf\{d(6,1), d(6,2)\}\}, \\
= \inf\{\inf\{4,3\}, \inf\{5,4\}\}, \\
= \inf\{3,4\} = 3, \ldots, \\
\sup(D(B_1, A)) = \sup\{D(B_1, A_1), D(B_1, A_2)\}, \\
= \sup\{3,1\} = 3, \ldots, \\
\nu(\mathcal{B}, \mathcal{A}) = \inf\{\sup_{A \in \mathcal{A}}(D(B_1, A)), \sup_{A \in \mathcal{A}}(D(B_2, A)), \sup_{A \in \mathcal{A}}(D(B_3, A)\}, \\
= \inf\{3,5,7\} = 3.
\end{aligned}$$

Let $\Im, \Omega$ denote the images in Fig. 20.10 and Fig. 20.11, respectively, and let $X = \Im \cup \Omega$. In the approach space $(X, D_{\rho_{\|\cdot\|}})$, consider the sample nearness measurements[4] in Table 20.1. ■

Fig. 20.14. Sample ROI $\mathcal{N}$

20.6 Regions-of-Interest in Z. Pawlak's Paintings

In this section, sample regions-of-interest (ROIs) are considered in terms of their nearness to each other. A *image region* such as the one in Fig. 20.14 is a collection $\mathcal{N} \in \mathcal{P}(X)$ of neighbourhoods of points in the interior of the region.

The distance measurement framework provided by ε-approach nearness considered in the context of ROIs in digital pictures lead to a number of different

[4] The measurements in Table 20.1 are from an implementation of $(X, D_{\rho_{\|\cdot\|}})$ created by H. Fashandi to handle comparison of pairs of visual neighbourhoods. Let $N_{\mathbf{x}} and N_{\mathbf{y}}$ denote a pair of visual image neighbourhoods, then use $D_{\rho_{\|\cdot\|}}(\{N_{\mathbf{x}}\}, \{N_{\mathbf{y}}\})$ to compute values like those in Table 20.1.

applications such as classification of fingerprints, authentication of written signatures, analysis of digital images from remote sensors, microfossils in oil exploration core samples, web page regions, and satellite images useful in detecting in planetary surface or stellar changes. In this section, the focus is on an application of ε-approach nearness in comparing regions in Z. Pawlak's paintings such as those shown in Fig. 20.15. Notice that $N_{\mathbf{x}} \in ROI_1, N_{\mathbf{y}} \in ROI_2$ in Fig. 20.15, where ROI_1, ROI_2 are collections of neighbourhoods. ε-approach nearness is ideally suited for a study of collections of neighbourhoods, since the assertion that *ROIs are near or apart* is made if, and only if, there is at least one pair of neighbourhoods that are sufficiently near or apart. The end result is that we consider all pairs of neighbourhoods in selected ROIs in assessing the nearness or apartness of the ROIs.

The answer to the question *How near are Pawlak's paintings?* can be answered fairly accurately by ε-approach nearness distances between descriptions of members of collections of neighbourhoods. To give nearness measurements more focus, small regions-of-interest (ROIs) (*e.g.* horizon, shoreline) are considered.

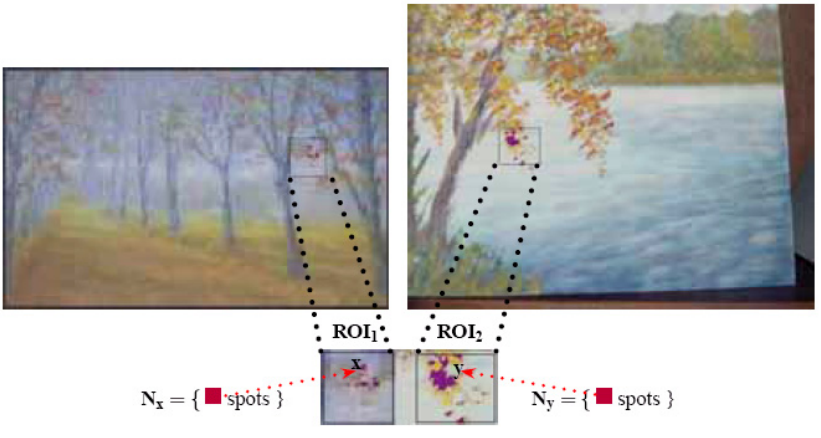


Fig. 20.15. Sample Nbd^s in ROIs

Example 20.11. **Sample ROIs in a Pawlak Paintings**
Sample ROIs in paintings of brightly-coloured autumn trees by Z. Pawlak are shown in Fig. 20.15. Assume that each ROI is selected manually. After identifying the ROIs, it is then possible to identify visual neighbourhoods of each point in the ROIs (see, *e.g.* $N_{\mathbf{x}} \in ROI_1, N_{\mathbf{y}} \in ROI_2$ in Fig. 20.15). To determine the nearnessor apartness of selected ROIs, it is necessary to identify the features of image pixels needed to extract the descriptions of the pixels in the form of feature vectors. Detection of patterns (repetition of shapes or colours or textures like those found in Pawlak's paintings) in a visual neighbourhood then reduces to determining the perceptual distances between neighbourhoods of ROI points. By perceptual distance, we mean

determining the distance $D_{\rho_{\|\cdot\|}}(\{N_{\mathbf{x}}\},\{N_{\mathbf{y}}\})$ for pairs of neighbourhoods $N_{\mathbf{x}}, N_{\mathbf{y}}$ for for all pairs of neighbourhoods. ■

Example 20.12. **Sample ROIs in Pawlak's Paintings**
Consider, first, watershadow ROIs in two of Pawlak's paintings. Sample watershadow ROIs are shown in Fig. 20.16. For example, the tip-of-the-shoreline ROI in Fig. 20.15 (repeated in Fig. 20.16.2) can be compared with the ROI shown in Fig. 20.16.1. Other Pawlak watershadows are shown in Fig. 20.16.3 and Fig. 20.16.4. In addition, watershadows are shown in Fig. 20.16.5 and Fig. 20.16.6. ■

Table 20.2. Comparison Between Pawlak Waterscapes

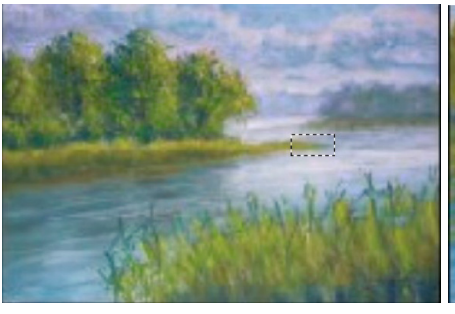
feature(s)	Pawlak Autumn ROI vs. Spring ROI	ε-value	$\nu_{D_{\rho_{\|\cdot\|}}}$-value
Edge	shoreline ROI (Fig. 20.16.1 vs. Fig. 20.16.2)	20	0.23
Intensity	shoreline ROI (Fig. 20.16.1 vs. Fig. 20.16.2)	4	3.00
Intensity, Edge	shoreline ROI (Fig. 20.16.1 vs. Fig. 20.16.2)	33	4.97
Edge	trees ROI (Fig. 20.16.3 vs. Fig. 20.16.4)	54	13.12
Intensity	trees ROI (Fig. 20.16.3 vs. Fig. 20.16.4)	4	3.00
Intensity, Edge	trees ROI (Fig. 20.16.3 vs. Fig. 20.16.4)	54	50.12
Edge	shadows ROI (Fig. 20.16.5 vs. Fig. 20.16.6)	2	1.08
Intensity	shadows ROI (Fig. 20.16.5 vs. Fig. 20.16.6)	15	14.00
Intensity, Edge	shadows ROI (Fig. 20.16.5 vs. Fig. 20.16.6)	16	15.08

Example 20.13. **Sample ROIs in Pawlak & Monet Waterscapes**
In this example, a waterscape painted in 1954 by Pawlak is compared with a waterscape painted by C. Monet in 1929. The APm toolset[5] can be used to select image ROIs. This is the first step in determining the nearness of ROIs. Sample ROIs are given in Fig. 20.16 and Fig. 20.17. Probe functions for two features are currently available in the APm, namely, pixel gradient and intensity (more will be added later). ROI neighbourhoods are determined after pixel features have been selected. ■

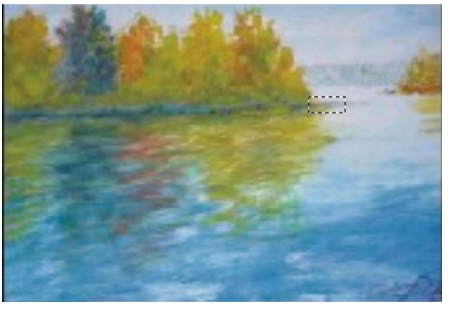
Example 20.14. **Sample ε-Approach Nearness Measurements**
A selection of ε approach merotopic nearness measurements is given in Table 20.2 and Table 20.3. In each case, a ε value is given in the case where the merotopy $\nu_{D_{\rho_{\|\cdot\|}}}(\mathcal{A},\mathcal{B}) < \varepsilon$. Pixel gradient (*i.e.* edge feature in the measurement tables) consistently requires a low ε value to obtain a merotopy measurement that indicates that

[5] Available at `http://wren.ece.umanitoba.ca`. This distribution of APm includes not only the tool itself but also a report that explains how to use the tool to select and compare digital image ROIs. Nearness measurements are carried out using an approach merotopy.

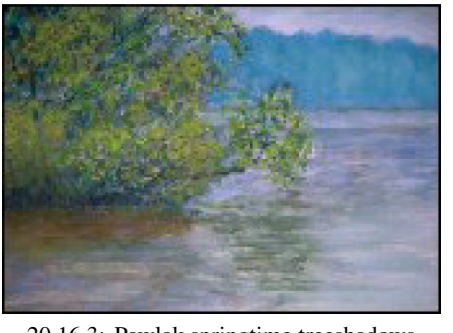
the collections of neighbourhoods in a pair of ROIs are near. From a pixel gradient perspective, the selected shoreline and building ROIs in the sample paintings by Pawlak and Monet provide an indicator of the similarities of the painting techniques of both artists. The results in these measurement tables are inconclusive. Many more ROIs of varying size should be considered to arrive at an assessment of the nearness of the patterns represented by the ROIs. ■



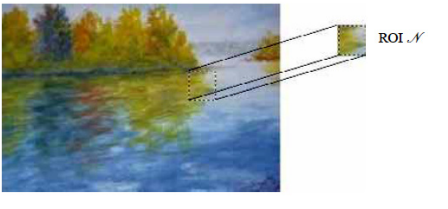
20.16.1: Pawlak springtime shoreline



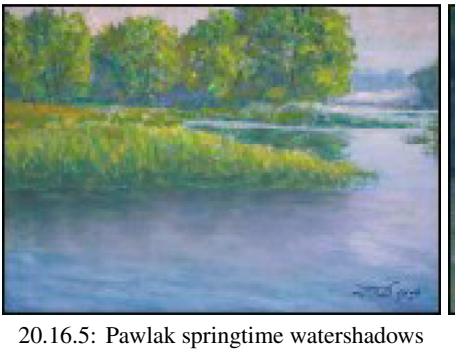
20.16.2: Pawlak fall shoreline



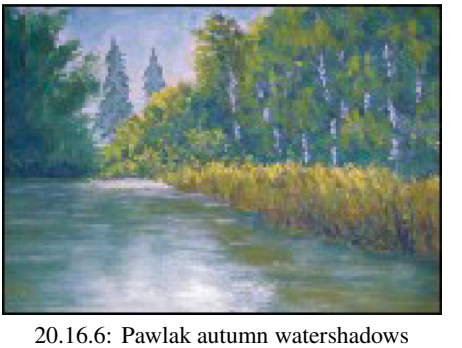
20.16.3: Pawlak springtime treeshadows



20.16.4: Pawlak autumn treeshadows



20.16.5: Pawlak springtime watershadows

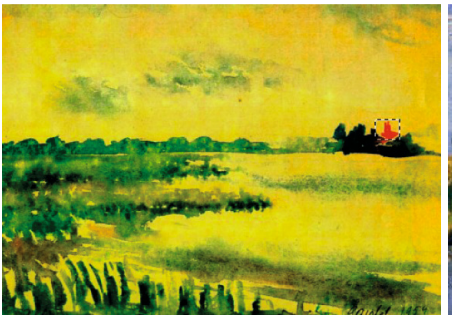


20.16.6: Pawlak autumn watershadows

Fig. 20.16. Comparing Pawlak Waterscapes

20.17.1: Pawlak shoreline 1954

20.17.2: Monet shoreline 1929



20.17.3: Pawlak building 1954

20.17.4: Monet building 1929

Fig. 20.17. Sample Measurements

Table 20.3. Comparison Between Pawlak and Monet Waterscapes ROIs

feature(s)	Pawlak ROI vs. Monet ROI	ε-value	$\nu_{D_{\rho_{\|\cdot\|}}}$-value
Edge	shoreline ROI (Fig. 20.17.1 vs. Fig. 20.17.2)	8	0.49
Intensity	shoreline ROI (Fig. 20.17.1 vs. Fig. 20.17.2)	20	1.00
Intensity, Edge	shoreline ROI (Fig. 20.17.1 vs. Fig. 20.17.2)	20	20.34
Edge	building ROI (Fig. 20.17.3 vs. Fig. 20.17.4)	7	6.12
Intensity	building ROI (Fig. 20.17.3 vs. Fig. 20.17.4)	20	28.00
Intensity, Edge	shoreline ROI (Fig. 20.17.3 vs. Fig. 20.17.4)	30	11.55

20.7 Concluding Remarks

This chapter offers a tribute to Zdzisław Pawlak's extraordinary talent as a painter. It focuses on the some of the subtleties in Pawlak's paintings that span over 50 years,

beginning in the early 1950s. These subtleties are especially evident in Pawlak's rendition of Springtime and Autumn skies and watershadows cast by lake shoreline trees and marshland reeds and cattails.

In considering the patterns in Pawlak's paintings, this chapter introduces a visual perception approach in digital image analysis (briefly, *perceptual image analysis*). This approach has been motivated by a need to solve the image correspondence problem in terms of perceived resemblances between digital images. What the eye sees should correspond, to some extent, to measures of nearness between pairs of images. Pointers on how to go about establishing a perceptual image analysis can be found in studies of representative space models of physical continua by Poincaré toward the end of the 19th century and the introduction of approach spaces by R. Lowen in 1989, elaborated in the years between 1997 and 2003.

In addition, the parallel discoveries about spatially near sets that began with F. Riesz in 1908 and continued with the introduction of proximity spaces in the seminal work by S.A. Naimpally in 1970, amplified and extended by others, led to the recent introduction of ε-approach nearness spaces inspired by M. Katětov's work on merotopology. This chapter is attentive to two research streams (*i.e.* approach space stream and merotopology stream) in presenting a viable approach to solving the image correspondence problem and in arriving at a satisfactory approach to measuring the nearness of Pawlak's paintings.

Acknowledgements. Many thanks to Som Naimpally, Surabhi Tiwari, Maciej Borkowski, Sheela Ramanna, Andrzej Skowron, Jarek Stepaniuk, Piotr Wasilewski, Christopher Henry, Homa Fashandi, Amir H. Meghdadi and Shabnam Shahfar for their suggestions and insights concerning topics in this chapter. I also want to thank Homa Fashandi for her work on implementing approach merotopies, Christopher Henry for the work he has done in putting together the NEAR system, a complete implementation of near sets tailored to the needs of image analysis that is reported in this chapter, and Maciej Borkowski for designing the APm system that implements approach merotopies using the new form of ε-approach merotopy. This research has been supported by the Natural Sciences & Engineering Research Council of Canada, *that is*, *Conseil de recherches en sciences naturelles et en génie du Canada* (NSERC) grant 185986, Manitoba Centre of Excellence Fund (MCEF) grant and the Canadian Arthritis Network grant SRI-BIO-05.

References

1. Beer, G., Lechnicki, A., Levi, S., Naimpally, S.: Distance functionals and suprema of hyperspace topologies. In: Annali di Matematica Pura Ed Applicata CLXII(IV), pp. 367–381 (1992)
2. Bourbaki, N.: Elements of Mathematics. General Topology, Pt. 1. Hermann, Addison-Wesley, Paris, Reading (1966, 1974)
3. Engelking, R.: General Topology, Revised & completed edition. Heldermann Verlag, Berlin (1989)
4. Hausdorff, F.: Grundzüge der Mengenlehre, pp. Viii+ 476. Veit and Company, Leipzig (1914)

5. Hausdorff, F.: Set Theory, p. 352. AMS Chelsea Publishing, Providence (1914)
6. Henry, C.: Near sets: Theory and applications, Ph.D. dissertation, supervisor: J.f. Peters. Ph.D. thesis, Department of Electrical & Computer Engineering (2010)
7. Henry, C., Ramanna, S.: Parallel Computation in Finding Near Neighbourhoods. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 523–532. Springer, Heidelberg (2011)
8. Hocking, J., Young, G.: Topology, Dover, NY (1988)
9. Katětov, M.: On continuity structures and spaces of mappings. Comment. Math. Univ. Carolinae 6, 257–278 (1965)
10. Khare, M., Tiwari, S.: Grill determined *L*-approach merotopological spaces. Fund. Inform. 48, 1–12 (2010)
11. Khare, M., Tiwari, S.: *L*-approach merotopies and their categorical perspective. International Journal of Mathematics and Mathematical Sciences 2010, 1–16 (2010), doi:10.1155/2010
12. Leader, S.: On clusters in proximity spaces. Fundamenta Mathematicae 47, 205–213 (1959)
13. Liu, G.: Closures and topological closures in quasi-discrete closure spaces. Appl. Math. Let. 23, 772–776 (2010)
14. Lowen, R.: Approach Spaces: The Missing Link in the Topology-Uniformity-Metric Triad, Oxford Mathematical Monographs, pp. Viii + 253. Oxford University Press, Oxford (1997)
15. Lowen, R., Vaughan, D., Sioen, M.: Completing quasi metric spaces: an alternative approach. Houstan J. Math. 29(1), 113–136 (2003)
16. Naimpally, S.: Proximity Spaces, pp. X + 128. Cambridge University Press, Cambridge (1970)
17. Naimpally, S.: Reflective functors via nearness. Fund. Math. 85, 245–255 (1974)
18. Naimpally, S.: Generalized metric spaces via annihilators. Questions and Answers in General Topology 9, 203–226 (1991)
19. Naimpally, S.: Plasticity in metric spaces. J. of Math. Analyis and Applications 313(1), 38–48 (2006)
20. Pawlak, Z.: Classification of objects by means of attributes. Institute of Computer Science, Polish Academy of Sciences, ICS PAS Reports, vol. 429, pp. 1-20. Warsaw (1981)
21. Pawlak, Z.: Rough sets. International J. Comp. Inform. Science 11, 341–356 (1981)
22. Pawlak, Z., Peters, J.: Jak blisko (how near). In: Systemy Wspomagania Decyzji, vol. I, pp. 57–109. University of Silesia, Katowice (2007) ISBN 83-920730-4-5
23. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177, 41–73 (2007)
24. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177, 28–40 (2007)
25. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
26. Peters, J.: Near sets. General theory about nearness of objects. Ap. Math. Sci. 1(53), 2609–2629 (2007)
27. Peters, J.: Near sets. Special theory about nearness of objects. Fund. Inf. 75(1-4), 407–433 (2007)
28. Peters, J.: Sufficiently near sets of neighbourhoods. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 17–24. Springer, Heidelberg (2011)
29. Peters, J.F., Borkowski, M.: ε-near collections. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 533–542. Springer, Heidelberg (2011)

30. Peters, J., Naimpally, S.: Approach spaces for near families. Gen. Math. Notes 2(1), 159–164 (2011)
31. Peters, J.F., Skowron, A.: Zdzisław Pawlak: Life and Work. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 1–24. Springer, Heidelberg (2006)
32. Peters, J., Skowron, A., Stepaniuk, J.: Nearness in approximation spaces. In: Proc. Concurrency, Specification and Programming (CS&P 2006), pp. 435–445. Humboldt Universität (2006)
33. Peters, J., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
34. Peters, J., Tiwari, S.: Approach merotopies and near filters. Theory and application. Gen. Math. Notes 3(1), 1–15 (2011)
35. Peters, J., Tiwari, S.: Completing extended metric spaces. An alternative approach. Applied Mathematics Letters (2012), doi:10.1016/j.aml.2012.01.012
36. Peters, J.F., Wasilewski, P.: Foundations of near sets. Inf. Sci. 179(18), 3091–3109 (2009), http://dx.doi.org/10.1016/j.ins.2009.04.018
37. Puzio, L., Peters, J.: Nearness of Subtly Different Digital Images. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 543–552. Springer, Heidelberg (2011)
38. Skowron, A., Peters, J.: Rough-granular computing. In: Handbook on Granular Computing. ch. 13, pp. 285–328. John Wiley & Sons Inc., USA (2008)
39. Thron, W.: Topological Spaces. Holt, Rinehart and Winston Inc., NY (1966)
40. Tiwari, S.: Some aspects of general topology and applications. Approach merotopic structures and applications, supervisor: M. khare. Ph.D. thesis, Department of Mathematics, Allahabad (U.P.), India (2010)
41. Čech, E.: Topological Spaces. John Wiley & Sons, NY (1966) revised Ed. by Z. Frolik and M. Katătov
42. Wang, L., Liu, X., Tian, X.: A Generalization of Near Set Model. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 553–558. Springer, Heidelberg (2011)
43. Wolski, M.: Gauges, Pregauges and Completions: Some Theoretical Aspects of Near and Rough Set Approaches to Data. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 559–568. Springer, Heidelberg (2011)

Chapter 21
An Implementation of the Zdzisław Pawlak Idea for Reasoning about Uncertainty: Approximate Reasoning by Parts

Lech Polkowski and Maria Semeniuk–Polkowska

Keywords: Rough sets, mereology, rough inclusions, concept approximations, many-valued mappings.

There is no inexact language for discoursing about inexact knowledge
(Zdzisław Pawlak)

21.1 Abstract

Professor Zdzisław Pawlak was the creator of Rough Set Theory, a paradigm for reasoning about uncertainty, whose underlying ideas were the idea that there are concepts which are wholly understandable (exact) and the idea of approximation of other concepts by exact ones. The theory of rough sets was and is written down in the language of naive set theory with objects of the real world as elements and concepts as their sets. In that setting, approximations are operators of interior and closure with respect to the topology induced by exact concepts. Our aim is to propose a higher–level language of mereology, that is, calculus of parts, in which concepts become elementary objects and relations among them are expressed as relations of being parts to degrees. This analysis allows in particular for approximations to various degrees which are introduced in this work.

This work is a homage to Professor Zdzisław Pawlak, memory of whom is still with us.

Lech Polkowski · Maria Semeniuk–Polkowska
Lech Polkowski, Polish–Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warszawa, Poland
e-mail: polkow@pjwstk.edu.pl

Maria Semeniuk–Polkowska, Chair of Formal Linguistics
Warsaw University, Browarna 8–12, 00-950 Warszawa, Poland
e-mail: m.polkowska@uw.edu.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 569–583.
springerlink.com

21.2 Introduction: The Language of Parts

Parts are the primitive notion of Mereology formulated in Leśniewski [4], cf., [5]. We introduce a binary predicate *part* with the meaning of the formula $part(u,v)$ that u is a part of v. The predicate *part* is to satisfy conditions,

(PT1) If $part(u,v)$ and $part(v,w)$ then $part(u,w)$,

(PT2) If $part(u,v)$ then it is not true that $part(v,u)$.

A weaker notion is that of an *ingredient* (due to T. Kotarbiński, see [4]), formally, the predicate *ingr* is defined as,

(INGR) $ingr(u,v)$ if and only if $part(u,v)$ or $u = v$.

These notions allow for some more precise relations among elements of the mereological universe. The predicate *ov* of *overlap* is defined as,

(OV) $ov(u,v)$ if and only if there is w such that $part(w,u)$ and $part(w,v)$.

Complementary to overlap is the predicate *dis* of *disjointness* defined as,

(DIS) $dis(u,v)$ if and only if it is not true that $ov(u,v)$.

We use the term *collection* in the intuitive sense equivalent to the term *property*; as we stay in the realm of a fixed collection of objects, no logical problems may arise. For a property F, one defines, cf. [4], the *class of* F, $Cls(F)$ by letting,

(CLS) (i) If $F(u)$ then $ingr(u,Cls(F))$ (ii) If $ingr(u,Cls(F))$ then some v,w satisfy $ingr(v,u)$, $ingr(v,w)$, $F(w)$.

Reasoning with parts rests with the Leśniewski Inference Rule,

(IR) Given u,v, if for each t such that $ingr(t,x)$ there exists z such that $ingr(z,t)$ and $ingr(z,y)$, then $ingr(x,y)$.

21.3 A Mereological Model for Rough Sets

Consider a collection F along with a sub–collection E, that is, $E(u)$ implies $F(u)$ for each u. We assume for simplicity, which does not exclude generality of exposition, that E consists of pair–wise disjoint elements, that is,

(DIS(E)) For u,v, if $E(u)$ and $E(v)$ then $dis(u,v)$.

A *mereological model for rough sets* is,

(MRS) $M_m = (F, E, part, ingr)$,

where *ingr* is an ingredient relation which satisfies along with the *part* relation condition (INGR). We call X an *exact concept* if and only if there exists a collection H which is a sub–collection of E such that $X = Cls(H)$.

We establish some properties of exact concepts.

Proposition 21.1. *Each e such that $E(e)$ is an exact concept.*

Proof. It remains to check that $e = Cls(e)$. Condition (i) in CLS is satisfied and condition (ii) follows by easily verified fact that e is the class of its ingredients. □

For exact concepts X, Y, with $X = Cls(H)$ and $Y = Cls(G)$, we define $X \vee Y = Cls(K)$, where $K(u)$ if and only if there exists e such that $E(e), H(e)$ or $G(e)$, $ingr(u, e)$. Similarly, $X \wedge Y = Cls(L)$, where $L(u)$ if and only if there exists e such that $E(e), H(e)$ and $G(e), ing(u, e)$, provided $ov(X, Y)$.

Proposition 21.2. *$X \vee Y$ and $X \wedge Y$ are exact concepts.*

Proof. We rely on the definition CLS of classes; we consider $X \vee Y$ first. Property (i) of CLS is satisfied manifestly. For property (ii), we consider v such that $ingr(v, X \vee Y)$. There are w, t such that $ingr(w, v), ingr(w, t)$ and there exists e such that $E(e), H(e)$ or $G(e), ingr(t, e)$. It follows by transitivity of *ingr* that $ingr(w, e)$ hence by definition CLS, $X \vee Y = Cls(P)$, where $P(e)$ if and only if $H(e)$ or $G(e)$, that is, $X \vee Y$ is an exact concept. In case of $X \wedge Y$ the proof goes along similar lines. □

It is known, cf., [4], that the equivalence $H \Leftrightarrow G$ implies $Cls(H) = Cls(G)$ and the implication $H \Rightarrow G$ implies $ingr(Cls(H), Cls(G))$. In virtue of this facts, logical tautologies $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$, $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$, $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$, $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$, and $p \vee q \Leftrightarrow q \vee p$, $p \wedge q \Leftrightarrow q \wedge p$ imply the following proposition.

Proposition 21.3. *For exact concepts X, Y, Z, the following identities hold,*

(i) $(X \vee Y) \vee Z = X \vee (Y \vee Z)$,

(ii) $(X \wedge Y) \wedge Z = X \wedge (Y \wedge Z)$,

(iii) $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$,

(iv) $X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$,

(v) $X \vee Y = Y \vee X$,

(vi) $X \wedge Y = Y \wedge X$. □

For an exact concept $X = Cls(H)$, we define its *complement* $-X$ as $-X = Cls(-H)$, where $-H$ denotes the collection of those e such that $E(e)$ which are not in H, that is, they satisfy $dis(e,X)$ in virtue of our assumption that elements in E are pair–wise disjoint. Logical tautologies $\neg(p \vee q) \Leftrightarrow (\neg p) \wedge (\neg q)$, $\neg(p \wedge q) \Leftrightarrow (\neg p) \vee (\neg q)$, and $\neg(\neg p) \Leftrightarrow p$ imply

Proposition 21.4. *For exact concepts X, Y, the following hold,*

(i) $-(-X) = X$,

(ii) $-(X \vee Y) = (-X) \wedge (-Y)$,

(iii) $-(X \wedge Y) = (-X) \vee (-Y)$. □

As *ingr* is a partial order, the notion of the supremum of a collection of exact concepts is in order. For a collection $\mathcal{Z}$=$(X_a : a \text{ in } A)$ an exact concept X is the supremum of $\mathcal{Z}$ if and only if $ingr(X_a, X)$ for each a in A and each exact concept Y such that $ingr(X_a, Y)$ for each a in A satisfies $ingr(Y, X)$.

Proposition 21.5. *For a collection H such that $H(e)$ if and only if there exists a in A such that $H_a(e)$, where H_a is a collection for each a in A, we have that $X = Cls(H) = sup_a X_a$, where $X_a = Cls(H_a)$.*

Proof. As $H_a(e) \Rightarrow H(e)$ it follows that $ingr(X_a, X)$ for each a in A. Assume that an exact concept Y is such that $ingr(X_a, Y)$. The inclusion $ingr(X, Y)$ follows by a direct application of the inference rule IR to the pair X, Y. □

Propositions 21.3, 21.4, 21.5 imply that

Proposition 21.6. *The collection of exact concepts with operations $\vee, \wedge, -$ is made into a complete Boolean algebra with the unit $Cls(E)$ and without the null element.* □

Let us stress here that the fact that each mereological universe carries a structure of a complete Boolean algebra without the null element was proved first in Tarski [14], where the definitions of operations $\vee, \wedge, -$ used above were given; this fact may not be understood as the statement that each mereological universe **is** merely the universe of a complete Boolean algebra without the null element, cf., Clay [1].

Proposition 21.6 generalizes the rough set result, cf., Pawlak [8] that exact sets generated by the partition by indiscernibility relation $IND(B)$ for a set B of attributes, form a complete Boolean algebra under operations $\cup, \cap, -$; in case the part relation is the strict containment $\subset$, and then the ingredient relation is the containment $\subseteq$, the class operator becomes the union $\bigcup$ of a family of sets, and Proposition 21.6 gives the rough set result when E is the set of indiscernibility classes; the only difference is the lack of the empty concept.

21.4 Approximations

For a concept X, in F, we define its *lower–E–approximation* $\underline{X}^E$ as follows,

(LA) $\underline{X}^E = Cls(L_X)$,

where $L_X(u)$ if and only if $E(u)$ and $ingr(u,X)$.

From this definition, the following properties of the lower–E–approximation follow,

Proposition 21.7. *The operator* $\underline{.}^E$ *satisfies,*

(i) Each concept $\underline{X}^E$ *is exact,*

(ii) If X *is an exact concept, then* $X=\underline{X}^E$,

(iii) $ingr(\underline{X}^E, X)$,

(iv) $\underline{\underline{X}^E}^E = \underline{X}^E$,

(v) If $ingr(X,Y)$ *then* $ingr(\underline{X}^E, \underline{Y}^E)$.

Proof. (i) follows by definition of an exact concept, (ii), (iv) follow directly from (i), (iii), (v) are obtained by the inference rule (IR). □

The *upper–E–approximation* $\overline{X}^E$ is defined as

(UA) $\overline{X}^E = Cls(U_X) = -Cls(D_X)$,

where $U_X(u)$ if and only if $E(u)$ and $ov(u,X)$, and, $D_X(u)$ if and only if $E(u)$ and $dis(u,X)$.

Proposition 21.8. *As with lower approximations, the upper–E–approximation operator satisfies,*

(i) Each concept $\overline{X}^E$ *is exact,*

(ii) If X *is an exact concept, then* $X=\overline{X}^E$,

(iii) $ingr(X, \overline{X}^E)$,

(iv) $\overline{\overline{X}^E}^E = \overline{X}^E$,

(v) If $ingr(X,Y)$ *then* $ingr(\overline{X}^E, \overline{Y}^E)$. □

In case when *ingr* is the containment $\subseteq$ on sets, in which case the class operator is the union of a family of sets, and E is the family of indiscernibility classes

of an indiscernibility relation $IND(B)$ for some set B of attributes, operators of lower– and upper–E–approximations become operators of B–lower– and B–upper–approximations of the rough set theory, cf., [8].

21.5 An Extension to Parts to a Degree: Rough Mereology

Rough mereology, cf., [9], is a theory of the notion of a part to a degree. It preserves mereology by the requirement that part to the greatest degree of 1 be the ingredient. The formal rendering of the idea of a part to a degree is by means of the predicate μ of rough inclusion. We use the same symbol μ for predicate and its interpretation as a relation in any model of mereology.

The predicate $\mu(x,y,r)$, under the generic name of *rough inclusion* [9] is introduced as satisfying in the model $(U, ingr)$ of mereology, the following conditions,

(RINC1) $\mu(x,y,1) \Leftrightarrow ingr(x,y)$,

(RINC2) $\mu(x,y,1) \Rightarrow \forall z[\mu(z,x,r) \Rightarrow \mu(z,y,r)]$,

(RINC3) $\mu(x,y,r) \wedge s < r \Rightarrow \mu(x,y,s)$.

The immediate consequences of postulates RINC1–RINC3, obtained directly or by means of the inference rule (IR), are

(RINC4) $\mu(x,x,1)$,

(RINC5) $\mu(x,y,1) \wedge \mu(y,z,1) \Rightarrow \mu(x,z,1)$,

(RINC6) $\mu(x,y,1) \wedge \mu(y,x,1) \Leftrightarrow x = y$,

(RINC7) $x \neq y \Rightarrow \neg\mu(x,y,1) \vee \neg\mu(y,x,1)$,

(RINC8) $\forall z \forall r[\mu(z,x,r) \Leftrightarrow \mu(z,y,r)] \Rightarrow x = y$.

By a *model* for rough mereology, we mean a quadruple,

$$(MRM)\ M = (V_M, \pi_M, ingr_M, \mu_M),$$

where V_M is a set with a part relation $\pi_M \subseteq V_M \times V_M$, the associated ingredient relation $ingr_M \subseteq V_M \times V_M$, and a rough inclusion $\mu_M \subseteq V_M \times V_M \times [0,1]$ which satisfies RINC1–RINC3.

A *rough mereological model for rough sets* is,

$$(RMRS)\ M_{rm} = (F, E, \mu, ingr),$$

where F, E are as in MRS (sect.) 21.3, μ is a rough inclusion, and *ingr* is an ingredient relation related to μ by means of (RINC1).

We restrict ourselves to two cases of rough inclusions, in [9] the reader may find a more detailed discussion. Both cases are related to continuous t–norms.

We recall that a t–norm T is a function $T : [0,1]^2 \to [0,1]$, which satisfies the following postulates,

indext–norm (TN1) T is *associative*: $T(T(x,y),z) = T(x,T(y,z))$,

(TN2) T is *commutative*: $T(x,y) = T(y,x)$,

(TN3) T is *non–decreasing in each coordinate*: $T(z,y) \geq T(x,y)$ *whenever* $z \geq x$,

(TN4) $T(1,x) = x$,

(TN5) $T(x,0) = 0$,

Moreover, T may satisfy an additional postulate,

(TN6) T *is continuous*.

'Classical' t–norms,

the Łukasiewicz t–norm $L(x,y) = max\{0, x+y-1\}$,

the Product t–norm $P(x,y) = x \cdot y$,

the Minimum t–norm $M(x,y) = min\{x,y\}$,

satisfy TN1–TN6. Let us observe that L and P satisfy,

(TN7) $T(x,x) < x$ *for each* $x \in (0,1)$.

A t–norm $T(x,y)$ which satisfies postulates TN6, TN7 is said to be an *Archimedean t–norm*.

Given a t–norm T, the *residual implication* $\Rightarrow_T$ is defined via the condition

$$(RI)\ x \Rightarrow_T y \geq z \Leftrightarrow T(x,z) \leq y \tag{21.1}$$

For further reference, we include here a useful list of well–known properties of residual implications.

Proposition 21.9. *For every t–norm T, the residual implication* $\Rightarrow_T$ *satisfies the following conditions,*

(RI1) $T(x,y) \leq z \Rightarrow_T u$ *if and only if* $x \leq T(y,z) \Rightarrow_T u$,

(RI2) $y \Rightarrow_T (z \Rightarrow_T u) = T(y,z) \Rightarrow_T u$,

(RI3) $y \Rightarrow_T z = 1$ *if and only if* $y \leq z$,

(RI4) $x \leq y \Rightarrow_T u$ *if and only if* $y \leq x \Rightarrow_T u$,

(RI5) $y \leq (y \Rightarrow_T z) \Rightarrow_T z$,

(RI6) $T(x,y) \Rightarrow_T 0 = x \Rightarrow_T (y \Rightarrow_T 0)$,

(RI7) $x \Rightarrow_T (y \Rightarrow_T r) \geq z$ *if and only if* $(z \Rightarrow_T x) \Rightarrow_T r \geq y$,

(RI8) $x \Rightarrow_T 0 \geq 0$.

Proof. Properties (RI1)–(RI4) follow straightforwardly by definition. We comment a bit on the last four. As $y \Rightarrow_T z \leq y \Rightarrow_T z$, we have $T(y \Rightarrow_T z, y) \leq z$; hence, by commutativity of T, it follows that $T(y, y \Rightarrow_T z) \leq z$ which implies $y \leq (y \Rightarrow_T z) \Rightarrow_T z$, that is, (RI5) follows.

Associativity of T implies that $z \leq T(x,y) \Rightarrow_T 0$ if and only if $T(z, T(x,y)) \leq 0$, that is, $T(T(z,x),y) \leq 0$ so equivalently $T(z,x) \leq y \Rightarrow_T 0$ which is equivalent to $z \leq x \Rightarrow_T (y \Rightarrow_T 0)$. From the equivalence of the first and the last statements (RI6) follows. Property (RI7) is a paraphrase in terms of $\Rightarrow_T$ of associativity of T whereas (RI8) does express the property that $T(x,0) = 0$. □

We recall, cf., [9], that each continuous t–norm T defines a rough inclusion μ_T by means of,

(RES) $\mu_T(x,y,r)$ if and only if $x \Rightarrow_T y \geq r$.

We recall that $\mu(x,y,1)$ if and only if $x \leq y$ and that the t–norms L, M, P induce the following rough inclusions (we give values in cases when $x \geq y$)

$\mu_L(x,y,r)$ if and only if $min\{1, 1-x+y\} \geq r$,

$\mu_P(x,y,r)$ if and only if $\frac{y}{x} \geq r$,

$\mu_M(x,y,r)$ if and only if $y \geq r$.

We recall, cf., [9], Ch. 6, that those rough inclusions obey the transitivity law; we include a proof.

Proposition 21.10. *For each continuous t–norm* T*, the transitivity rule holds for the rough inclusion* μ_T*: if* $\mu_T(x,y,r), \mu_T(y,z,s)$*, then* $\mu_T(x,z,T(r,s))$.

Proof. $\mu_T(x,y,r)$ is equivalent to $T(x,r) \leq y$, and, $\mu_T(y,z,s)$ is equivalent to $T(y,s) \leq z$. By coordinate–wise monotonicity of T, it follows that $T(T(x,r),s) \leq z$, and, by associativity of T, one obtains $T(x,T(r,s)) \leq z$; hence, $\mu_T(x,z) \geq T(r,s)$. □

It is well–known, cf., *for example*, [2], that L, P are the only (up to an isomorphism) Archimedean rough inclusions; a theorem due to Ling [6] asserts that for an Archimedean t–norm a representation holds,

(AR) $T(x,y) = g(f(x)+f(y))$,

where f is a continuous decreasing function from [0,1] into [0,1], and g is the pseudo–inverse to f (cf., *for example*, [9], Ch. 6).

In case of L, $f(x) = 1-x = g(x)$ for $x \in [0,1]$. We recall, cf. [9], that an Archimedean T defines a rough inclusion μ^T as,

(ARI) $\mu^T(x,y,r) \Leftrightarrow g_T(|x-y|) \geq r$,

equivalently,

$$\mu^T(x,y,r) \Leftrightarrow |x-y| \leq f_T(r) \tag{21.2}$$

A specific recipe for μ^L is

$$\mu^L(x,y,r) \Leftrightarrow |x-y| \leq 1-r \tag{21.3}$$

The counterpart of Proposition 21.10 obeys for Archimedean rough inclusions,

Proposition 21.11. *For each Archimedean t–norm T, the transitivity rule is: if $\mu^T(x,y,r)$ and $\mu^T(y,z,s)$, then $\mu^T(x,z,T(r,s))$.*

Proof. Assume $\mu^T(x,y,r)$ and $\mu^T(y,z,s)$, that is, $|x-y| \leq f_T(r)$ and $|y-z| \leq f_T(s)$. Hence, $|x-z| \leq |x-y|+|y-z| \leq f_T(r)+f_T(s)$, and hence, $g_T(|x-z|) \geq g_T(f_T(r)+f_T(s)) = T(r,s)$, that is, $\mu^T(x,z,T(r,s))$. □

21.5.1 Rough Inclusions in Information and Decision Systems

We recall, cf., [9], that in order to define μ^T in the setting of an information or decision system (U,A,d), where U is a set of *objects*, A a set of *attributes*, and d is a *decision*, cf. [8], for each pair $u,v \in U$, we define the set,

(DISC) $DIS(u,v) = \{a \in A : a(u) \neq a(v)\}$.

The rough inclusion μ^T is defined, cf. loc. cit., by means of,

(MU) $\mu_I^T(u,v,r) \Leftrightarrow g_T(\frac{|DIS(u,v)|}{|A|}) \geq r$.

In case of the Łukasiewicz t–norm L, the rough inclusion μ_I^L is given by means of the formula,

(AIRI) $\mu_I^L(u,v,r) \Leftrightarrow 1 - \frac{|DIS(u,v)|}{|A|} \geq r$.

We introduce the set $IND(u,v) = A \setminus DIS(u,v)$. With its help, a new form is,

$$\mu_I^L(u,v,r) \Leftrightarrow \frac{|IND(u,v)|}{|A|} \geq r \tag{21.4}$$

We quote from [9]: 'The formula (21.4) witnesses that the reasoning based on the rough inclusion μ^L is the probabilistic one. At the same time, we have given a logical proof for formulas like (21.4) that are very frequently applied in Data Mining and Knowledge Discovery, also in rough set methods in those areas, see, *for example*, Kloesgen and Żytkow [3]. It also witnesses that μ_I^L is a generalization of indiscernibility relation to the relation of partial indiscernibility.'

The transitivity property holds in this case as well

Proposition 21.12. *If* $\mu_I^T(u,v,r)$ *and* $\mu_I^T(v,w,s)$ *then* $\mu_I^T(u,w,T(r,s))$.

We include a proof from [9].

Proof. We begin with the observation that

$$DIS(u,w) \subseteq DIS(u,v) \cup DIS(v,w), \tag{21.5}$$

hence,

$$\frac{|DIS(u,w)|}{|A|} \leq \frac{|DIS(u,v)|}{|A|} + \frac{|DIS(v,w)|}{|A|} \tag{21.6}$$

We let,

$$\begin{cases} g_T(\frac{|DIS(u,v)|}{|A|}) = r \\ g_T(\frac{|DIS(v,w)|}{|A|}) = s \\ g_T(\frac{|DIS(u,w)|}{|A|}) = t \end{cases} \tag{21.7}$$

Then,

$$\begin{cases} \frac{|DIS(u,v)|}{|A|} = f_T(r) \\ \frac{|DIS(v,w)|}{|A|} = f_T(s) \\ \frac{|DIS(u,w)|}{|A|} = f_T(t) \end{cases} \tag{21.8}$$

Finally, by (21.6),

$$f_T(t) \leq f_T(r) + f_T(s) \tag{21.9}$$

hence,

$$t = g_T(f_T(t)) \geq g_T(f_T(r) + f_T(s)) = T(r,s) \tag{21.10}$$

witnessing $\mu_I^T(u,w,T(r,s))$. This concludes the proof. □

21.6 Approximations to a Degree

We apply rough inclusions μ_T and μ^T, μ_I^T denoted jointly with the symbol μ, the two latter in case $T = L$, in order to define approximations to a degree of concepts in F. For such a concept X, and $r \in [0,1]$, we define the *r–lower–$\mathcal{E}$–approximation*, where $\mathcal{E}$ stands for the collection of exact concepts over E,

(LRA) $\underline{X}^{\mathcal{E},\mu,r} = Cls(R_X)$,

where $R_X(u)$ if and only if $\mathcal{E}(u)$ and $\mu(u,X,r)$.

Proposition 21.13. *The following properties may be established for this approximation,*

(i) In case of $\mu = \mu^L, \mu_I^L$, which are symmetric, $ingr(\underline{X}^{\mathcal{E},\mu,1}, X)$,

(ii) In case of $\mu = \mu^L, \mu_I^L$, $\underline{X}^{\mathcal{E},\mu,1} = \underline{X}^{E,1}$,

(iii) $s \leq r$ implies $ingr(\underline{X}^{\mathcal{E},\mu,r}, \underline{X}^{\mathcal{E},\mu,s})$,

(iv) In case of $\mu = \mu^L, \mu_I^L$, $ingr((\underline{X}^{\mathcal{E},\mu,r})^{\mathcal{E},\mu,r}, \underline{X}^{\mathcal{E},\mu,r})$.

Proof. In case (i), it suffices to verify (ii) thanks to Proposition 21.7(iii). For (ii), it is manifest that $ingr(\underline{X}^{E,1}, \underline{X}^{\mathcal{E},\mu,1})$ by definitions of the two. Concerning the converse, let $ingr(t, \underline{X}^{\mathcal{E},\mu,1})$; by the class definition, there are u,w such that $ingr(u,t)$, $ingr(u,w)$, and, $\mathcal{E}(w)$, $\mu(w,X,1)$. Thus, $\mu(u,w,1)$, $\mu(u,t,1)$, hence, $\mu(t,u,1)$ and by transitivity, $\mu(t,w,1)$, hence, $\mu(t,X,1)$. By the inference rule IR, $ingr(\underline{X}^{\mathcal{E},\mu,1}, \underline{X}^{E,\mu,1})$ follows.

Properties (iii) and (iv) follow on the same lines by a direct appeal to the inference rule (IR). □

Let us observe that usage of rough inclusions in approximations makes it possible to place ourselves in abstract symbolic representation spaces of objects and indiscernibility classes: an approximation needs not in general be a subset of the approximated concept, contrary to the standard rough set approach. Virtues of this approach have been demonstrated in applications to classifier synthesis, cf., *for example*, [12].

A dual definition of the *r–upper–$\mathcal{E}$–approximation* follows,

(UAR) $\overline{X}^{\mathcal{E},\mu,r} = Cls(U_X)$,

where $U_X(u)$ if and only if $\mathcal{E}(u)$ and $\mu(X,u,r)$.

Proposition 21.14. *The following properties of the upper–$\mathcal{E}$–approximation may be established similarly to those of the lower approximation,*

(i) If $s \leq r$ then $ingr(\overline{X}^{\mathcal{E},\mu,r}, \overline{X}^{\mathcal{E},\mu,s})$,

(ii) $\overline{X}^{\mathcal{E},\mu,r}$ is an exact concept,

(iii) If the extension property: if $E(u)$ and $ov(X,u)$ then there exists w such that $\mathcal{E}(w)$ and $ingr(X,w)$ holds, then $ingr(\overline{X}^{E,1}, \overline{X}^{\mathcal{E},\mu,1})$. □

21.7 A Characterization of μ_L

The t–norm of Łukasiewicz as well as the residual implication of Łukasiewicz were characterized by Menu and Pavelka [7] in topological terms. In order to state their results, we recall necessary topological notions.

Each rough inclusion $\mu(x,y,r)$ can be regarded as a set–valued mapping, its values r in a convex subset of the interval [0, 1] containing 0; the topology in question is therefore the topology of set–valued mappings. For those mappings a notion of continuity is fused from two notions of semi–continuity; a mapping $f : X \to 2^Y$ from a topological space X into the collection of closed subsets of a topological space Y is *upper–semi–continuous* (usc) (respectively, *lower–semi–continuous* (lsc)) if and only if for each open set $P \subseteq Y$, the set $f^{\leftarrow}(P) = \{x : f(x) \subseteq P\}$ is open in X (respectively, the set $f^{\cap}(P) = \{x : f(x) \cap P \neq \emptyset\}$ is open in X). The mapping f is *continuous* if and only if it is upper– and lower–semi–continuous.

As, clearly, every mapping $f : X \to Y$ is a set–valued mapping with singleton values, the above notions can be easily translated into semi–continuity notions for those ordinary mappings: the mapping $f : X \to Y$ is *upper–semi–continuous* (respectively, *lower–semi–continuous*) if and only if for every r, the set $\{x : f(x) \geq r\}$ is closed (respectively, for every r, the set $\{x : f(x) \leq r\}$ is closed).

The couple of results established by Menu and Pavelka are, cf. [7],

Proposition 21.15. *(Menu–Pavelka [7]) Any t – norm T is lower–semi–continuous; moreover, any associative and commutative function T with $T(0,x) = 0$, $T(1,x) = x$ which is lower–semi–continuous is a t–norm. In this case the residual implication $\Rightarrow_T$ is given by the condition $y \Rightarrow_T z = max\{x : T(x,y) \leq z\}$.* □

Proposition 21.16. *(Menu–Pavelka [7]) For every t–norm T, continuity of $\Rightarrow_T$ implies continuity of T. In this case, T is isomorphic to L.* □

We follow these line of research and we apply these idea to the rough inclusion μ_L, cf., [10]. The first statement sets conditions for a residual implication.

Proposition 21.17. *([10]) Each function $\phi(x,y) : [0,1]^2 \to [0,1]$ which is non–increasing in the first argument x, non–decreasing in the second argument y, and it does satisfy conditions (RI3), (RI4), (RI7), (RI8), is of the form $\Rightarrow_T$ for some t–norm T, and it is necessarily upper–semi–continuous.*

Proof. The function $T(x,y)$ defined by the duality (i) $T(x,y) \leq r \Leftrightarrow \phi(x,r) \geq y$ satisfies conditions T1–T5, as observed earlier, (RI4), (RI7) do express commutativity and associativity of T, and (RI3), (RI8) are responsible for boundary conditions for T. Hence, duality (i) qualifies ϕ as the residuum $\Rightarrow_T$ which is upper–semi–continuous by the same duality and lower–semi–continuity of T demonstrated in Proposition 21.15. □

As each rough inclusion μ_T takes as its values closed intervals of the form $[0,\overline{\mu}(x,y)]$, we use the symbol $\overline{\mu}(x,y) \ni r$ for the fact that $\mu(x,y,r)$.

Proposition 21.18. *([10]) Each function $\mu(x,y) : [0,1]^2 \to 2^{[0,1]}$, whose values are closed intervals $[0,\overline{\mu}(x,y)]$ and which satisfies conditions,*

(MI1) $\overline{\mu}(x,y) \ni 1$ if and only if $x \leq y$,

(MI2) $\overline{\mu}(y,u) \ni x$ if and only if $\overline{\mu}(x,u) \ni y$,

(MI3) $\overline{\mu}(x,\overline{\mu}(y,r)) \ni z$ if and only if $\overline{\mu}(\overline{\mu}(z,x),r) \ni y$,

(MI4) $\overline{\mu}(x,0) \ni 0$,

and is non–increasing in x and non–decreasing in y is of the form μ_T for some t–norm T.

Proof. We let $\phi(x,y) \geq r$ if and only if $\overline{\mu}(x,y) \ni r$; it is straightforward to verify that conditions (MI1)–(MI4) correspond to conditions (RI3), (RI4), (RI7), (RI8) for ϕ and accordingly, ϕ is of the form $\Rightarrow_T$ by Proposition 21.17, hence, μ is μ_T. □

Proposition 21.19. *([10]) The set $E_T(s) = \{(x,y) \in [0,1]^2 : \overline{\mu}(x,y) < s\}$ is open for each s, that is, μ_T is upper–semi–continuous as a many–valued mapping.*

Proof. It follows by upper–semi–continuity of μ as a single-valued map. □

Proposition 21.16 has a counterpart for rough inclusions.

Proposition 21.20. *([10]) Each rough inclusion μ_T continuous, as a multi–valued mapping is isomorphic to μ_L in the sense that T is isomorphic to L.*

Proof. We consider the corresponding residuum $x \Rightarrow_T y = \overline{\mu}(x,y)$ and prove its continuity. Let $\overline{\mu}(x,y) \in U$ for an open set U; let $\overline{\mu}(x,y) \in (a,b) \subseteq U$.

Let $V = [0,a] \cup (a,b)$, and consider $W = \{(x,y) : [0,\overline{\mu}(x,y)] \subseteq V$ and $[0,\overline{\mu}(x,y)] \cap (a,b) \neq \emptyset\}$; clearly, W is open by continuity of μ_T and $(x,y) \in W$ if and only if $x \Rightarrow_T y \in U$, hence, $x \Rightarrow_T y$ is continuous and Proposition 21.16 implies that T is isomorphic to L. □

21.8 Acknowledgement

With this work, the authors acknowledge the influence of the late Professor Zdzisław Pawlak on their scientific careers. They remember his wit, intelligence, energy, enthusiasm not only in science, where he was one of leading figures, but also in Art and many other areas of activity.

21.9 Conclusions

The approximations described in this work have the primary feature that they are induced in abstract spaces of concept representations allowing for a greater flexibility in their definitions and, a fortiori, for possibly better application results. The ideas presented in this work of mereology have found applications, among others, in classification of data (classifier synthesis) and intelligent robotics. For some descriptions of these applications, the reader will be pleased to consult [11], [12], [13].

References

1. Clay, R.: Relation of Leśniewski's Mereology to Boolean Algebra. The Journal of Symbolic Logic 39, 638–648 (1974)
2. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
3. Kloesgen, W., Żytkow, J. (eds.): Handbook of Data Mining and Knowledge Discovery. Oxford University Press, Oxford (2002)
4. Leśniewski, S.: Podstawy Ogólnej Teoryi Mnogości, I (Foundations of General Set Theory, I, in Polish). Prace Polskiego Koła Naukowego w Moskwie, Sekcya Matematyczno–przyrodnicza (2)(1916)
5. Leśniewski, S.: In: Srzednicki, J., Surma, S.J., Barnett, D., Rickey, V.F. (eds.) Collected Works of Stanisław Leśniewski. Kluwer, Dordrecht (1992)
6. Ling, C.-H.: Representation of associative functions. Publ. Math. Debrecen 12, 189–212 (1965)
7. Menu, J., Pavelka, J.: A note on tensor products on the unit interval. Comm. Univ. Carolinae 17, 71–83 (1976)
8. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
9. Polkowski, L.: Approximate Reasoning by Parts. An Introduction to Rough Mereology. Springer, Berlin (2011)

10. Polkowski, L., Semeniuk–Polkowska, M.: Some remarks on rough inclusions. In: Szczuka, M., Czaja, L., Skowron, A., Kacprzak, M. (eds.) Proceedings of the XXth International Workshop on Concurrency, Specification and Programming (CS&P 2011), Pułtusk, Poland, pp. 408–415. Białystok University of Technology, Poland (2011)
11. Polkowski, L.: Granulation of Knowledge: Similarity Based Approach in Information and Decision Systems. In: Meyers, R.A. (ed.) Springer Encyclopedia of Complexity and System Sciences, Springer, Berlin (2009); article 00 788
12. Polkowski, L., Artiemjew, P.: On Granular Rough Computing: Factoring Classifiers Through Granulated Decision Systems. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 280–289. Springer, Heidelberg (2007)
13. Polkowski, L.: Ośmiałowski, P.: Spatial reasoning with applications to mobile robotics. In: Jing, X.-J. (ed.) Motion Planning for Mobile Robots: New Advances, pp. 433–453. InTech, Vienna (2008)
14. Tarski, A.: Zur Grundlegung der Booleschen Algebra. I. Fundamenta Mathematicae 24, 177–198 (1935)

Chapter 22
Granular Concept Mapping and Applications

Sumalee Sonamthiang, Kanlaya Naruedomkul, and Nick Cercone

Abstract. This chapter presents a granular concept hierarchy (GCH) construction and mapping of the hierarchy for granular knowledge. A GCH is comprised of multilevel granular concepts with their hierarchy relations. A rough set based approach is proposed to induce the approximation of a domain concept hierarchy of an information system. A sequence of attribute subsets is selected to partition a granularity, hierarchically. In each level of granulation, reducts and core are applied to retain the specific concepts of a granule whereas common attributes are applied to exclude the common knowledge and generate a more general concept. A granule description language and granule measurements are proposed to enable mapping for an appropriate granular concept that represents sufficient knowledge so solve problem at hand. Applications of GCH are demonstrated through learning of higher order decision rules.

Keywords: Information granules, granular knowledge, granular concept hierarchy, granular knowledge mapping, granule description language, higher-order rules, multilevel partitioning, attribute selection

22.1 Introduction

An *information system* in a rough set paradigm [10], [11] is a basic knowledge representation method in an attribute-value system. An information system is represented in a table in which a row keeps an object and each column keeps the

Sumalee Sonamthiang
Institute for Innovative Learning, Mahidol University, Nakhon Pathom, Thailand 73170
e-mail: tew197@gmail.com

Kanlaya Naruedomkul
Mathematics Department, Faculty of Science, Mahidol University, Bangkok, Thailand 10400
e-mail: scknr@mahidol.ac.th

Nick Cercone
Department of Computer Science and Engineering, Faculty of Science and Engineering,
York University, Toronto, Ontario, Canada M3J 1P3
e-mail: ncercone@yorku.ca

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 585–603.
springerlink.com

value of the corresponding attribute. The tabular representation simplifies recording the objects into an information system, especially in real-time transactions by capturing a transaction separately and using a single global representation for every record in every situation. However, an occurrence of a transaction may be related to other transactions in the problem space. Representation in this fashion is seen as a flat and unconnected structure that hides the meaningful relations in the data.

An information analysis based on an attribute-value system considers the values of the attribute subsets to extract relationships within data. The relationships can be classified into two types: internal and external. An internal relationship is the relation between attributes' values within a single object, whereas an external relationship provides connections between many objects. Classical rule discovering methods extract internal relationships from a decision table; however, the obtained rules represent fragmented knowledge and hide the meaningful relationships among objects of a universe. An example of an internal relation rule is expressed by: $IF < x,a >= v_{a1} THEN < x,d >= v_{d_1}$. The rules obtained from internal relationships can be superfluous. Postprocessing is necessary to reduce rules' conflicts, shorten the rule premise, shrink the size of the rule set, or group together similar rules. The rules obtained from internal relations represent fragmented knowledge and remain embodied in hidden meaningful relationships among objects in a universe. Postprocessing to improve quality of the rules has been studied, *for example*, evaluation of association rules' importance [9] and mining higher-order decision rules [26].

Unlike internal relationships, an external relationship among objects does not only provide knowledge of higher-order rules but also for concept approximation.

Table 22.1. Some examples of animal data set

label	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	...	class
aardvark	1	0	0	1	0	0	1	1	...	1
antelope	1	0	0	1	0	0	0	1	...	1
bass	0	0	1	0	0	1	1	1	...	4
bear	1	0	0	1	0	0	1	1	...	1
boar	1	0	0	1	0	0	1	1	...	1
buffalo	1	0	0	1	0	0	0	1	...	1
calf	1	0	0	1	0	0	0	1	...	1
carp	0	0	1	0	0	1	0	1	...	4
catfish	0	0	1	0	0	1	1	1	...	4
cavy	1	0	0	1	0	0	0	1	...	1
cheetah	1	0	0	1	0	0	1	1	...	1
chicken	0	1	1	0	1	0	0	0	...	2
clam	0	0	1	0	0	0	1	0	...	7
crab	0	0	1	0	0	1	1	0	...	7
crayfish	0	0	1	0	0	1	1	0	...	7
crow	0	1	1	0	1	0	1	0	...	2

For example, given animal data in the information system in Table 22.1, a human categorizes and conceptualizes a concept differently.

One may give a name of concept represented by class 1 as mammal class without concerning surrounding data. But for a machine, it can learn that only the attribute value *milk=1* is sufficient to determine the class 1 precisely (based on given data sets). In this example, however, some attributes' values such as *egg=0*, *hair=1*, *toothed*=1 are correlated with (but not necessary dependent) *milk=1*. These features can be seen as dominate attribute subset in which together the attribute subset has more gravity to draw animal class abstraction. Therefore, a machine can form a concept by using the most dominant attribute subset on the decision class to mimic human granular conceptualization. On the other hand, if one is asked to differentiate animal in class 1, one needs to granulate knowledge relative to more of the detailed features such as size, domestic/wild, and legs based on given data.

Various attribute subsets can be considered to obtain external relations from different dimensions; thus, groups of related objects can be discovered. The knowledge obtained from this type of relationship is represented as clusters attached with each clusters' description[4],[5], [20]. Moreover, the relationships among the objects can be local or global; specifically, relations can be extracted in many levels of granularity. We hypothesize that discovering external relationships between objects in a universe can be used to approximate the connections of objects and form multilevel granular concepts.

Rough set theory (RST) [10], [11] provides a formal framework that focuses on both internal and external relations. For extensions on rough sets please refer to [12], [13], [14]. In rough sets, the indiscernibility relation expresses the external relations between objects and the relation can be used to form a granular concepts. Rough sets also influence Granular Computing (GrC), an emerging paradigm for computing of concept approximation [1], [15]. *A granular concept represents sufficient information to solve a problem at hand. How coarse or how specific should a granular concept be to convey such sufficient information?*

In this study, a granular concept hierarchy (GCH) and granular concept mapping are presented. GCH is a multilevel of granularity of a domain knowledge in hierarchical structure. This structure provides rich information for a problem solver and mapping mechanism to search for an appropriate level of granularity. GCH comprises of a root node, a set of nonroot nodes, a non empty set of leaves, and the hierarchy relations. A node in a tree can be seen as a granule in which instances in the node hold similar properties to a certain degree, and they are part of their parent. Thus, a parent holds the common properties of its children, and the siblings have a certain degree of similarity to each other by the common properties.

We present two algorithms to construct a GCH. The first algorithm is to recursively partition an information system into a GCH. The second algorithm computes the selection of a sequence of attribute subsets which is necessary to partition a granularity hierarchically. *Common attributes* (defined as the subset of attributes that forms indiscernibility relations among the objects of a granule) and the attributes' values are united to form the granular concept's description. At each level of granulation, reducts and core are applied to retain the specific concepts of a gran-

ule, whereas common attributes are applied to exclude the common knowledge and generate a more general concept. We also present a granule description language that provides semantic encoding as well as an interpretation of which semantics a granule concept conveys. The semantics are encoded by rough set approximation. Degree of coarseness/specificity of a granular concept, then, can be interpreted for a target concept.

The chapter is outlined as follows: In the next section, related works in hierarchical information granulation are explored. A formal definition of a GCH and an example are given in Section 22.3. Section 22.4 details two algorithms to construct a GCH hierarchically. Section 22.5 reports our evaluation and results of higher-order rules learning from a GCH of an artificial Zoo database. Finally, Section 22.6 presents conclusion and discussion of possible extensions.

22.2 Related Study

In this section, previous studies on multilevel granular concept approximation are reviewed. There are various approaches to approximate uncertain concepts from uncertain data. Four main approaches are focused which are rough sets, fuzzy sets, near sets, and shadowed sets.

The fuzzy sets and shadowed sets provide contributions to GrC [1] for information processing by using continuous membership grades induction [2], [16],[17],[27]. Hierarchical fuzzy sets and shadowed sets can be identified by further refinement of the sets. Multilevel fuzzy sets can be approximated based on previous layer of fuzzy sets in order to obtain multilevel granular concepts. Therefore, defuzzification is needed to map for granular fuzzy concept indexing. It is preferable if the approximated concept mechanism provide descriptive knowledge and knowledge evaluation for hierarchical granular mapping.

RST was proposed by Zdzisław Pawlak (1926-2006) in 1982. The theory is to model indiscernible (similar) objects and forms a basic granule of knowledge about a domain, based on given observations (see, *e.g.*,[18],[21]). However, the observations can be imperfect: inconsistent, insufficient and uncertain. These characteristics of observations, consequently, cause basic granules being rough which are defined as rough sets. Defining rough sets does not require priori probabilistic information about data. Moreover, the rough sets permits induction of rules about uncertainty [6], namely the certain classes as certain rules, and the uncertain classes as possible rules.

In [19], James F. Peters proposed a special theory of near sets. Near sets are disjoint sets that resemble each other to a certain degree. Resemblance of near sets can be obtained using a probe function such as closeness (qualitatively near) between objects as well as other probe functions that return values of object features such as color, shape, texture, and duration. The closeness is determined by the objects' features. Near sets is an extension of rough sets with nearness function for granular concept approximation. A granular knowledge approximation based on near sets is

to identify family of an instance x to an instance y by *link relation*. Therefore, a granule X and a granule Y are near sets to each other if and only if mapping the link relation of x with y is *sufficiently* large. The author also proposed a framework to enable searching for relevant nearness relation relative to the problem being solved through a distance measurement. Based on the rough sets, we proposed an idea of domination attribute subset partitioning to granulate an information granule into lower level granules. Like the near set approach, using domination attributes has advantages in linking a family of an object together. We also apply reduct and core attributes to retain specific information until the lowest level of granulation.

Hoa and Son [7] introduced a complex concept approximation approach based on a layered learning method together with RST. The authors used taxonomy as the domain knowledge and attribute values in the data_set to guide composing attributes into intermediate concepts until the target concept is obtained. The target concepts are the concepts in the decision attribute. However, the domain taxonomies are usually unavailable to guide the layer learning and need to be discovered before applying this approach.

A study of granularity-based formal concepts is presented in [21], [25], for example. In [25], the authors defined a formal concept by a pair consisting of its intension and extension $(\phi, m(\phi))$, where ϕ is a logical rule of a subset of attributes with the attributes' values and $m(\phi)$ is a granule obtained by partitioning the universe of objects using the attribute subset ϕ. Moreover, Yao [24] presented an approach to hierarchical granulation based on rough sets called stratified rough approximation. The stratified rough set approximation is a simple multi-level granulation based on nesting of one-level granulation (*e.g.*, granulation by the equivalence relation). Yao [24] presented three methods for multi-layered granulation which are as follows:

- nested rough set approximations induced by a nested sequence of equivalence relations,
- stratified rough set approximations induced by hierarchies, and
- stratified rough set approximations induced by neighborhood systems.

In the nested granulation approach, the granulation starts with indiscernibility relations on a set of objects represented by attribute-value vectors. Then the subsequent indiscernibility relations are defined by successively removing attributes from the set of remaining attributes. Sequencing of attribute subsets for partitioning is determined by dependencies between condition attributes. The sequence of attribute subsets for partitioning affects a granules' extension and the hierarchy structure. By this approach, the obtained hierarchy structures are predefined by the attributes' dependencies. Thus, the approach is unconcerned about the objects similarities which are very important in the sense of clusters. Moreover, there are some information systems that have no attribute dependency. In the stratified rough set approximations induced by hierarchies, levels of hierarchies provide the sequence of granulation. As mentioned earlier, the hierarchy of a domain may be unavailable. In [23], the authors also described the use of neighborhood systems to induce hierarchial partitioning. The neighborhood system $NS(x)$ is a nested family of subsets of the universe, with each neighborhood representing a specific level of similarity to x.

However, an information system can contain a huge number of attributes. The issue of attribute subset selection for measuring the closeness, similarity or proximity in an information system is not studied.

Yao [24] recommended a motivating idea for our approach, that is, stratified approximation can be used to search for an appropriate level of accuracy for an application. Therefore, a map of granular concepts which provides rich information about domain structure is developed.

22.3 Granular Concept Hierarchy

In this section, we shall formally define elements and the hierarchy structure of our granular concept mapping approach. Definitions of a GCH with its syntax and semantics are given and detailed in the next subsection. Then, the granular knowledge evaluation is also presented through the semantics of the target concept approximation using a rough set-based approach.

22.3.1 Formal Definitions of a Granular Concept Hierarchy

A GCH is a hierarchical granular knowledge organization that provides multilevel granular knowledge units, evaluation of knowledge, and knowledge mapping mechanism.

A hierarchy of granular concept mapping is formally defined as a quadruple

$$GCH = < G, R, T, \alpha >, \tag{22.1}$$

where G is a non-empty set of *nodes*, and the nodes themselves are non-empty set. R denotes a relationship between two nodes. T denotes the target concept of a node, and α denotes the accuracy approximation of the target concept T .

R is a binary relation of *parent-child* and *child-parent* relation on g. If $< g, g' > \in R$ then g is the parent of g' and g' is a child of g. There is a designated element r of G called *root*. r holds the universe of elements such that $\neg r = \emptyset$. A branch $BR = g_0, g_1, g_2, ..., g_n$ is the maximal sequence of element of G such that $g_0 = r$, and for every $i \geq 0, < g_i, g_{i+1} > \in R$. Nodes g which $R(g) = \emptyset$ are called *leaves*. The level of g, denoted by $\|g\|$, is defined by n if and only if there is a branch $BR = g_0, g_1, g_2, ..., g_n$, where $g = g_n$. Obviously, $\|r\| = 0$.

T is the target concept of granule which is defined by a set of decision attribute values.

α is knowledge evaluation of a granule g. The knowledge evaluation in our approach is defined by accuracy of rough approximation:

$$\alpha(g) = \frac{|LOWER(g)|}{|UPPER(g)|}. \tag{22.2}$$

$LOWER(g)$ is lower approximation and $|UPPER(g)|$ is upper approximation of a granule g induced by a subset of attribute. $|X|$ denotes the cardinality of a set X. The approximation accuracy is in the range of $0 \leq \alpha(X) \leq 1$, and $\alpha(\emptyset) = 1$.

A GCH comprises of nodes in which the coarsest concept is represented at the root level, whereas the most specific concept is represented at the leaf levels. We articulate a concept by using the idea of the most dominant attribute subset: the more dominant degree attribute subset, the more gravity to draw the objects into concepts by that subset. Once a concept is granulated by the most dominant attributes subset, we obtain the more specific concepts which are drawn by *common attribute subset*. The common attribute subset forms the indiscernibility relations among the concept's extension. This structure allows mapping of appropriate granular knowledge in order to solve a problem at hand. The essences of GCH knowledge organization are as follows.

- In order to map to an appropriate granular knowledge, the problem solver must identify satisfaction criterions. One of satisfaction criterion is that the granular knowledge is evaluated by *sufficient knowledge* for solving a particular problem. If the problem is to find decision rules to predict unseen objects, then the appropriate levels of granularity can be found in the granules which no children of them have smaller boundary regions. If the problem is to predict missing values of condition attributes of an object, then the appropriate levels of granularity can be found at the leaf levels where the objects are indiscernible. One may define a satisfaction criterion by setting precision tolerance of applying the granular knowledge. This criterion permits reducing cost of computation where precisiation is expensive or unavailable.
- Because GCH provides multilevel of granular knowledge ranging from the coarsest level at the root and the most specific level at the leaves, GCH structure provides system of granular knowledge mapping through a tree traversal. Searching for a granular concept in a GCH can be achieved through several techniques such as the depth first search and breadth first search.
- Core attributes are essential to form the more specific concepts since they contains specific characteristics of an object. In GCH construction, core attributes are preserved to retain such specific concepts until the latest granulation.

We shall define the syntax and semantics of GCH and present algorithms to construct a GCH as follows.

22.3.2 Syntax and Semantics of a Granular Concept

This section explains what knowledge is represented in the granular concepts and how to interpret and evaluate knowledge in a granular concept. The section is started by definitions of basic notions, followed by syntax and semantics of a granular concept.

Definition 22.1. *Let g be a node in a map of granular concepts M and g is a decision table, $g \subseteq D$. A common attribute of g is the attribute that forms the indiscernibility relation on $g \times g$. The set of common attributes is denoted by $CA, CA \subseteq A$.*

Definition 22.2. *The set of target concepts of g, denoted by $\tau(g)$, is defined by the set of decision values in the decision attribute of $x \in g$.*

$$\tau(g) = \bigcup v_d | < x, d >= v_d, \forall x \in g. \tag{22.3}$$

Definition 22.3. *The most dominant target concept, $\hat{\tau}$, is defined by the decision value of the largest decision class in g.*

Definition 22.4. *A granular concept description phrase of g, denoted by $\pi(g)$, comprises of atomic predicates. A predicate is defined by a pair of a common attribute's name and a value of the attribute. Each predicate is conjuncted by the $\wedge$ operator to form a phrase.*

$$\pi(g) = ca_0(V_{ca_0}) \wedge ca_1(V_{ca_1}) \wedge ca_2(V_{ca_2}) \wedge ... \wedge ca_n(V_{ca_n}), \tag{22.4}$$

where $ca_i \in CA$ and $|CA| = n, 1 \leq i \leq n$.

Definition 22.5. *A granular concept description language of g is denoted by $\lambda(g)$. The language $\lambda(g)$ is generated by traversing M from g_0 to g. The phrases of the traversed granules are $\wedge$ conjuncted successively to form $\lambda(g)$.*

$$\lambda(g) = \pi(g_0) \wedge \pi(g_1) \wedge ... \wedge \pi(g). \tag{22.5}$$

Definition 22.6. *Syntax of a granular concept g is denoted by a pair:*

$$\psi =< \phi(g), \lambda(g) > \text{ if and only if } x \models \lambda(g), \forall x \in g. \tag{22.6}$$

$\phi(g) = \{x | x \in g\}$ is called concept's extensions and every member of $\phi(g)$ is understood by $\lambda(g)$. Note that $\lambda(g)$ is the granular concept's intension.

Definition 22.7. *Semantics of a granular concept is the accuracy of rough approximation of the granule toward the most dominant target concepts and the concept's intension. The semantics of g is denoted by $\xi(g)$*

$$\xi(g) = \frac{|LOWER(g)|}{|UPPER(g)|}, \tag{22.7}$$

where

$$LOWER(X) = \bigcup [g]_B | x \in g, [g]_B \subseteq g,$$

$$UPPER(X) = \bigcup [g]_B | x \in g, [g]_B \cap g \neq \emptyset,$$

$$[g]_B = \bigcup \{[< a, v >] | a \in B, B = CA \cup \{d\}\}, f(x,a) = \hat{\tau}.$$

A granular concept is indexed by its intension. The granular concept conveys a semantic of being a target concept ($\hat{\tau}.$). To interpret the concept's semantic, one can measure the rough approximation accuracy toward the target concept based on the granular concept intension. Example 1 provides an illustrative explanation.

Example 1. The decision table of Flu diagnosis in Table 22.2 contains four condition attributes of symptoms $\{Temperature, Headache, Nausea, Cough\}$ and one decision attribute $\{Flu\}$.

Table 22.2. Flu diagnosis

Cases	Temperature	Headache	Nausea	Cough	Flu
1	high	yes	no	yes	yes
2	very high	yes	yes	no	yes
3	high	no	no	no	no
4	high	yes	yes	yes	yes
5	normal	yes	no	no	no
6	normal	no	yes	yes	no

There are six cases of patient. If the first partitioning is $\{Headache\}*$, two granular concepts of g_1 and g_2 are obtained as shown in Table 22.3. If the equivalence relation is used to discern patients, there is one *common attribute* $CA = \{Headache\}$ for both g_1 and g_2. The description language is $\lambda(g_1) = Headache(yes)$ and $\lambda(g_2) = Headache(no)$. The target concept of g_1 is *having Flu*, and the semantics conveyed by g_1 is *the patients who have headache also get flu*, with the accuracy of approximation is 3/4. For g_2, the target concept is *having no Flu*. The semantics of *having no Flu* is 2/2 of the patients who have no headache.

Table 22.3. Granulated concepts of Flu diagnosis

$g_1 : \lambda(g_1) = Headache(yes)$					
Cases	Temperature	Headache	Nausea	Cough	Flu
1	high	**yes**	no	yes	yes
2	very high	**yes**	yes	no	yes
4	high	**yes**	yes	yes	yes
5	normal	**yes**	no	no	no
$g_2 : \lambda(g_2) = Headache(no)$					
Cases	Temperature	Headache	Nausea	Cough	Flu
3	high	**no**	no	no	no
6	normal	**no**	yes	yes	no

By the definitions, granular concepts can be approximated and interpreted to obtain their semantic. The next section gives details of GCH construction. A recursive

partitioning algorithm is proposed as well as an attribute subset selection algorithm to partition the granularity hierarchically.

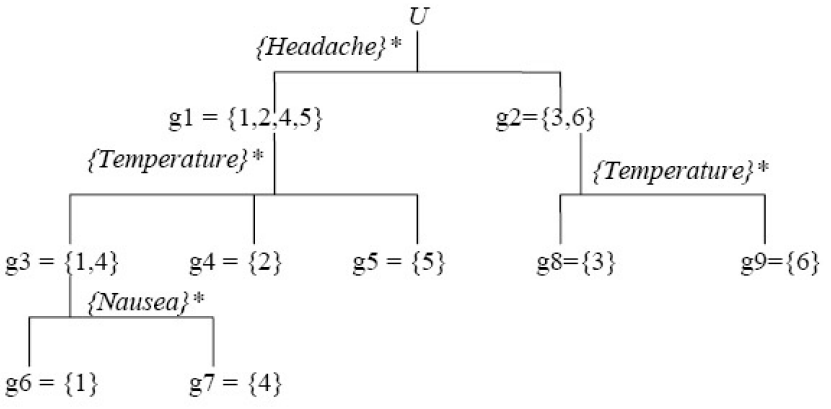


Fig. 22.1. A granular concept hierarchy for the Flu case base

22.4 Granular Concept Hierarchy Construction

The GCH construction is a recursive granulation in top-down manner. Specifically, the recursive construction is given in Section 22.4.1, where the symbols are the ones defined in Section 22.3. We also present an algorithm for attribute subset selection based on the most dominant degree of the attribute subset as illustrated in Section 22.4.2. In Section 22.4.3, the mapping for an appropriate granularity on the GCH is detailed.

22.4.1 An Algorithm for Recursive Granulations

Given a data set as an information system format, all observations start in one cluster, and granulations are performed recursively as one moves down the hierarchy. We design a recursive hierarchy construction as given in Algorithm 22.1. The input to this algorithm is an information system and the output is a GCH. The process begins with finding common attribute subset. Then, a temporary decision (*TempD*) table is derived from the current decision table by removing the common attributes. The *TempD* is not necessary if there is no common attribute. The attribute sequencing is accomplished through local attributes subset selection in the recursive partitioning. We select the most dominant attribute subset based on the attributes' values available in the decision table. We determine the domination using Algorithm 22.2.

Algorithm 22.1. Granular Concept Hierarchy Construction

Input : a decision table, $D =< g,(A \cup \{d\}),(V_a)_{a \subset A},V_d,f >$.
Output: a granular concept map, $M =< G,R >$.
$g \longleftarrow D$.
$g_0 \longleftarrow g$. // the root of the hierarchy
$TempD \longleftarrow D$.
$B \longleftarrow \emptyset$. // attribute subset for partitioning
$CA \longleftarrow \emptyset$. // the set of common attributes
$granulatedStatus(g) = false$. //mark the granulated concepts
Function GCHconstruct(g)
begin
  if *(g is discernible)* **then**
    1. Find common attribute subset CA of g;
    2. Generate a granule description phrase //See Definition 22.4.
    3. **if** $(CA \neq \emptyset)$ **then**
      $A \longleftarrow A - AC$
      $TempD \longleftarrow < g,(A \cup \{d\}),(V_a)_{a \subset A},V_d,f >$
    4. $B \longleftarrow MostDAselect(TempD)$.
    //Select the most dominant attribute subset B, see Algorithm 22.2.
    5. Partition the $TempD$ by B, $\{B\}* = g_1,g_2,...,g_n$.
    6. Generate relations of $< g,g_1 >,< g,g_2 >,...,< g,g_n > \in R$.
    for *all* $g_i, < g,g_i > \in R$ **do**
      7. Find $\hat{\tau}$, the most dominant target concept of g_i.
      // see Definition 22.3.
      8. Compute semantics of g_i.
      // see Definition 22.7.
      9. $granulatedStatus(g_i) \longleftarrow false$.
    10. $granulatedStatus(g) \longleftarrow true$; //mark g as granulated.
    11. $g \longleftarrow g_1$.
    12. $TempD =< g,(A \cup \{d\}),(V_a)_{a \subset A},V_d,f >$
  else
    Make a leaf granule.
end
for *all* g_i, $(R(g_i) = R(g))$ *and* $(granulatedStatus(g_i) = false)$ **do**
  GCHconstruct(g_i).

The selected attributes subset is then used to partition *TempD* and assign relationships between the obtained granules (children) and the original granule (parent). If a granule cannot be partitioned by the indiscernibility relation, a leaf node is generated.

22.4.2 An Algorithm for Level-Wise Attribute Selection

In this section, we present an algorithm for attribute subset selection which the selected attribute subset is used in partitioning a granule at each level by

Algorithm 22.2. Most Dominant Attribute Subset Selection

Input : a decision table ($TempD$), CORE, parameter N and ε.
Output: The most dominant attribute subset B // this attribute subset will be used to partition the $TempD$ in Algorithm 22.1.

$MostDA \longleftarrow \emptyset$.
$TopDA \longleftarrow \emptyset$.
$B \longleftarrow \emptyset$.
for *each* $a \in A$ **do**
  for *each* $v_a \in V_a$ **do**
    1. $[x]_{a \cup d} \longleftarrow \bigcup\{[a,v] | f(x,a) = v_a, f(x,d) = v_d\}$
  2. $domDegree(a_i) \longleftarrow \arg\max(|[x]_{a \cup d}|)$
$MostDA \longleftarrow \arg\max(domDegree(a_i))$.
$TopDA \longleftarrow TopN \arg\max(domDegree(a_i))$
$B \longleftarrow MostDA$
for *each* $TopB$, $TopB \subseteq TopDA$ *and* $|TopB| > 1$ **do**
  $[x]_{TopB} \longleftarrow \bigcup\{[a,v] | f(x,a) = v_a, a \in TopB\}$
  if $\arg\max(|[x]_{topB}|) > \varepsilon$ **then**
    $B \longleftarrow topB$
if $B - CORE \neq \emptyset$ **then**
  $B \longleftarrow B - CORE$
Return B

Algorithm 22.1. Algorithm 22.2 is designed to compute the most dominant attribute subset selection.

The rough set exploration system (RSES version 2.2) [3] is used to calculate reducts of the universe. Then *CORE* can be derived from intersection of all reducts. Given a decision table (temporary), we find the N most dominant attributes toward the decision class. *CORE* is used to preserve the specific feature(s) of instances in the granule by retaining *CORE* until the latest granulations. N can be tuned up to the number of condition attributes to compose a concept. In other words, our algorithm allows a flexible number of attributes in a subset for partitioning. We use co-occurrence counting of attributes' values and decision classes to determine the domination degree. Once the most N dominant attributes are obtained, we determine the co-occurrences within the N attributes to find if any combination of them can be used to approximate a concept by threshold ε. A count of co-occurrence among condition attributes' values implies the degree of which these attribute values can be used to compose a common concept. We tune the ε by the number of instances in working granule. The subset of attributes with the greatest domination degree, and the greatest domination degree is greater than the threshold, is selected to partition the current granule. If no domination degree of the N combination attributes meets the threshold ε, the single most dominant attribute is selected.

Example 2 illustrates the recursive construction of a GCH using Algorithm 22.1 and Algorithm 22.2.

Example 2. A GCH construction for the Flu diagnosis decision table (Table 22.1) is described step by step. The granulation starts by partitioning the universe (Table 22.1). In this example, the equivalence relation is used. The size of attribute subset to partition is one ($N = 1$) since the number of condition attributes is relatively small. The objects in the universe are discernible by the equivalence relation. Thus, we find reducts for this table which are, $\{Temperature, Headache, Nausea\}$, $\{Temperature, Nausea, Cough\}$, and $\{Headache, Nausea, Cough\}$, and core is $\{Nausea\}$. There is no common attribute value in this granule. We select the first attribute subset by determining the degree of attribute dominations. $Headache$ has the highest domination degree ($domDegree = 3$) compared with the rest of the condition attributes ($domDegree = 2$). Thus, the first attribute subset to partition is $\{Headache\}$ and $g_1 = \{1,2,4,5\}$ and $g_2 = \{3,6\}$ are obtained. Then we continue granulate g_1 selecting the most dominant attributes for g_1. $Temperature, Nausea$ and $Cough$ attributes have the same degree of domination. $Nausea$ is the core; thus, it is retained at this granulation. We can select $Temperature$ or $Cough$ to partition g_1. If we apply $Temperature$, we obtain granule $g_3 = \{1,4\}$, $g_4 = \{5\}$, $g_5 = \{2\}$ which are children of g_1. The granule g_4 and g_5 are indiscernible so they are leaf granule. We then granulate g_3 by finding common attribute subset which is $\{Cough\}$. The $Cough$ attribute can be now removed. The remaining attribute $\{Nausea\}$ is then used to partition g_3 to obtain $g_6 = \{2\}, g_7 = \{3\}$. Since all siblings are now leaf nodes we can return to the higher levels. We continue granulate g_2. Note that the temporary table can be generated as the common attribute $\{Headache\}$ is removed. Like partitioning g_1, $Nausea$ is retained. If we partition g_2 by $Temperature$, the indiscernible granule $g_8 = \{3\}$ and $g_9 = \{6\}$ are obtained. Fig. 22.1 shows the GCH for the Flu diagnosis domain.

22.4.3 Mapping for Appropriate Granularity in a GCH

Our approach of GCH does not only provide a multilevel of granular concept representation of variables, but also enables searching and evaluating techniques for a granular variable. In order to solve a problem, an application can perform a search in the GCH for an *appropriate* level of granularity. As a result, the *appropriate* level of granularity is evaluated by sufficient knowledge for solving a particular problem. If the problem is to obtain decision rules to diagnose unseen objects, then the appropriate levels of granularity can be found in the granules which no children of them have smaller boundary regions. If the problem is to predict missing values of condition attributes of an object, then the appropriate levels of granularity can be found at the leaf levels where the objects are indiscernible. Searching for a granular concept in a GCH can be achieved through several techniques such as the depth first search and breadth first search.

Table 22.4. Higher-order rules for the Zoo database

Rules	Accuracy	Coverage
Rule 1: IF animals have the same values in condition attribute $\{C,E,J,K\}$ THEN they are in the same class of $\{R\}$	0.956	0.979
Rule 2: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{B,D,F,H,I,M,O,P\}$ THEN they are in the same class of $\{R\}$	0.972	0.209
Rule 3: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{D,L,M,P\}$ and animals have the same values in condition attributes $\{B,F,G,I,N,O\}$ THEN they are in the same class of $\{R\}$	1.0	0.266
Rule 4: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{D,P\}$ and animals have the same values in condition attributes $\{G,I,M,Q\}$ THEN they are in the same class of $\{R\}$	0.961	0.293
Rule 5: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{F\}$ and animals have the same values in condition attributes $\{B,D,M,O\}$ THEN they are in the same class of $\{R\}$	0.988	0.562
Rule 6: IF animals have the same values in condition attributes $\{C,E,J,K\}$ and animals have the same values in condition attributes $\{B,D,I,L,M,N,O\}$ THEN they are in the same class of $\{R\}$	0.998	0.534

G1: {squirrel, fruitbat, vampire, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, dolphin, porpoise, wallaby, sealion, platypus, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl, calf, goat, pony, reindeer, pussycat}
- G11: {fruitbat, vampire}
- G12: {platypus}
- G13: {squirrel, hare, vole, mole, opossum, cavy, hamster seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer elephant ,giraffe, oryx , boar ,cheetah ,leopard ,lion ,lynx ,mongoose, polecat, puma ,raccoon, wolf, mink, girl, dolphin, porpoise, calf, goat, pony, reindeer}
  - G131: {dolphin, porpoise}
  - G132: {squirrel, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl calf, goat, pony, reindeer}
    - G1321: {squirrel, gorilla, wallaby, girl, hare, vole, cavy, hamster, antelope, buffalo, deer, elephant, giraffe, oryx, calf, goat, pony, reindeer, mole, opossum, aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose polecat, puma, raccoon, wolf, pussycat}
      - G13211: {hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx, mole, opossum aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
        - G132111: {mole, opossum, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
        - G132112: {aardvark, bear}
        - G132113:{hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx}
      - G13212: {squirrel, gorilla, wallaby}
      - G13213: {girl}
      - G13214: {cavy, hamster, calf, goat, pony, reindeer, pussycat}
        - G1321411: {cavy}
        - G1321412: {hamster calf goat pony reindeer}
        - G1321413: {pussycat}
    - G1322: {seal, sealion}
    - G1323: {mink}

G2: {chicken, dove, duck, lark, parakeet, pheasant, sparrow, wren, kiwi, crow, gull, hawk, skimmer, skua, ostrich, flamingo, swan, penguin, rhea, vulture}
- G21: {lark, pheasant, sparrow, wren, duck, kiwi, crow, hawk, gull, skimmer, skua}
  - G211: {kiwi}
  - G212: {lark, pheasant, sparrow, wren, duck, crow, hawk, gull, skimmer, skua}
    - G2121: {lark, pheasant, sparrow, wren, crow, hawk}
      - G21211: {lark, pheasant, sparrow, wren}
      - G21212: {crow, hawk}
    - G2122: {duck, gull, skimmer, skua}
      - G21221: {duck}
      - G21222: {gull, skimmer, skua}
- G22: {chicken, dove, parakeet}
- G23: {ostrich, flamingo, swan, rhea, vulture, penguin}
  - G231: {ostrich, flamingo, rhea, vulture}
    - G2311: {ostrich, rhea}
      - G23111: {ostrich}
      - G23112: {rhea}
    - G2312: {flamingo, vulture}
      - G23121: {flamingo}
      - G23122: {vulture}
  - G232: {swan penguin}
    - G2321: {swan}
    - G2322: {penguin}

G3: {pitviper, seasnake, slowworm, tortoise, tuatara, carp, haddock, seahorse, sole, bass, catfish, chub, dogfish, herring, pike, piranha, stingray, tuna, frog, frog, newt, toad}
- G31: {bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna}
  - G311: {carp, haddock, seahorse, sole}
  - G312: {bass, catfish, chub, herring, piranha, dogfish, pike, tuna}
    - G3121: {bass, catfish, chub, herring, piranha}
    - G3122: {dogfish, pike, tuna}
  - G313: {stingray}
- G32: {pitviper, slowworm, tortoise, tuatara}
- G33: {seasnake}
- G34: {frog, frog, newt, toad}
  - G341: {frog, newt}
    - G3411: {frog}
    - G3412: {newt}
  - G342: {frog}
  - G343: {toad}

G4: {flea, termite, gnat, honeybee, housefly, ladybird, moth, wasp, clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish}
- G41: {clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish, flea, termite}
  - G411: {seawasp, starfish, flea, termite, slug, worm, crab, crayfish, lobster, clam}
    - G4111: {flea, termite, slug, worm}
      - G41111: {flea, termite}
      - G41112: {slug worm}
    - G4112: {seawasp}
    - G4113: {starfish, crab, crayfish, lobster, clam}
      - G41131: {clam}
      - G41132: {starfish, crab, crayfish, lobster}
        - G411321: {crab}
        - G411322: {starfish}
        - G411321: {crayfish, lobster}
  - G412: {scorpion}
  - G413: {octopus}
- G42: {gnat, ladybird}
  - G421: {gnat}
  - G422: {ladybirdl}
- G43: {housefly, moth, wasp}
  - G431: {housefly, moth}
  - G432: {wasp}
- G44: {honeybee}

Fig. 22.2. A granular concept hierarchy for the Zoo data set

22.5 Evaluation

We evaluate the usefulness of a GCH through higher-order decision rules learning. The definition of higher-order rules are introduced by Yao [26]. A higher order rule expresses connections of different objects based on their attribute values. An example of a higher-order rule is "*if object x is related to object y with respect to an attribute set a, then x is related to y to another attribute set b.*" Yao recommends a mining of higher-order rules from a transformed decision table, where an entity is a pair of objects from the original table. However, transforming the n objects table generates $\frac{n!}{((n-2)!*2!)}$ pairs of objects. We present an alternative approach to extract higher-order decision rules from a GCH where no transformation process is required.

The data set used in the experiment is the Zoo database from the UCI machine learning repository. This database contains 101 objects, 17 condition attributes and one decision attribute. The condition attributes include 16 boolean-valued attributes and a numerical attribute. The decision attribute contains 7 classes of animal type. There is no missing value in this datas et. We construct a GCH for the Zoo data set as shown in Fig 22.2.

There can be several groups of animals that hold the same attributes' values in a subset of condition attributes. For example, there are 6 groups of animals clustered by attribute set $\{C,E,J,K\}$ which are Feathers, Milk, Backbone, Breathes respectively. These attributes draw a concept of *mammal* when $C=0,E=1,J=1$, and $K=1$. The concept of *bird* is drawn when $C=1,E=0,J=1$, and $K=1$, the concept of *amphibia* is formed by $C=0,E=0,J=1$, and $K=1$. The *arthropod* (bug) concept is formed by $C=0,E=0,J=0$, and $K=1$. The concept of *fish* is formed by $C=0,E=0,J=1$, and $K=0$. The concept of being *crustacean* is formed when $C=0,E=0,J=0$, and $K=0$. Note that, these groups will be granulated until all the member of the group are indiscernible. The concept descriptions of the animal groups are used to generate the higher-order rules. Once the hierarchy is obtained, a depth first tree search is performed to find the maximum level of accuracy of each branch. The higher-order rules are obtained from conjunctive connection of granular concepts' intensions along the visited branches. The extracted higher-order decision rule set for the Zoo data base is given in the first column in Table 22.3. Number of conjunction shows the level of hierarchy, starting from level 0 at the root. The higher-order rules are applied to the total of 5,050 pairs of animals, and there are 1,177 pairs of animals that belong to the same class. We measure the rules' accuracy and coverage which were used by [22] as follows:

$$accuracy(premise \Rightarrow conclusion) = \frac{|\phi(premise \wedge conclusion)|}{|\phi(premise)|}, \tag{22.8}$$

$$coverage(premise \Rightarrow conclusion) = \frac{|\phi(premise \wedge conclusion)|}{|\phi(conclusion)|}, \tag{22.9}$$

where $\phi(g)$ is the granule's extension, and $|x|$ denotes the cardinality of the set x. The results of the rules' accuracy and coverage are shown in the second and the third column of Table 22.4, respectively.

We shall discuss the interestingness of the higher-order decision rules as follows. The higher-order rule is the type of knowledge in more abstract level. This knowledge should be firstly applied to solve a problem. Naturally, given two animals, one can differentiate them by the concepts, not by the detailed of each attribute value if not necessary. The higher-order rules provide the concepts upon the domain which the rules can be applied for only some groups. The rules obtained from our approach have much higher accuracy degree than the coverage degree. This is because of the tree traversal searches for the maximum accurate level of each branch, where their children do not have smaller boundary regions than the parents. Once the target granules are found, the granules' language can be used to express the connections between objects in the same granule directly. The connections are multi-dimension which reflect the relationships between attributes in the attribute subset (*e.g.*, dependencies) and also the relationships between the condition attribute subset and the decision attribute. On the other hand, if one prefers the rule with higher coverage degree, the bread first search for the coarser granules can be achieved.

22.6 Conclusion

An approach to automatically construct a GCH from a decision table is presented. A GCH represents knowledge in different level of specificness/coarseness. A granular concept is formally defined for its syntax, semantic, and interpretation. With this rich information, an application can map from a granular concept that conveys sufficient information to solve a problem. The usefulness of the GCH is shown from the ability to extract higher-order rules from the GCH structure without postprocessing required. Extensions of this work include granular concept mapping based on a conceptual network of a domain for real world applications such as the educational and instructional area.

References

1. Bargiela, A., Pedrycz, W.: Granular Computing. An introduction. Kluwer Academic Publishers, Dordrecht (2003)
2. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human-centered information processing. IEEE Transactions on Fuzzy Systems 16(2), 320–330 (2008)
3. Bazan, J.G., Szczuka, M.S.: The Rough Set Exploration System. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)

4. Biswas, G., Weinberg, J.B., Fisher, D.H.: ITERATE: a conceptual clustering algorithm for data mining. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 28(2), 219–230 (1998)
5. Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis. LNCS (LNAI), vol. 3626. Springer, Heidelberg (2005)
6. Grzymała-Busse, J.W., Siddhaye, S.: Rough set approaches to rule induction from incomplete data. In: Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2004, Perugia, Italy, July 4-9, vol. 2, pp. 923–930 (2004)
7. Hoa, N.S., Son, N.H.: Rough set approach to approximation of concepts from taxonomy. In: Proc. of Knowledge Discovery and Ontologies Workshop, KDO 2004 at ECML/PKDD 2004, pp. 13–24 (2004)
8. Hu, X., Cercone, N.: Learning in relational databases: A rough set approach. J. Computational Intelligence 2, 323–337 (1995)
9. Li, J., Cercone, N.J.: A Rough Set Based Model to Rank the Importance of Association Rules. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005, Part II. LNCS (LNAI), vol. 3642, pp. 109–118. Springer, Heidelberg (2005)
10. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
11. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
12. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
13. Pawlak, Z., Skowron, A.: Rough sets: some extensions. Information Sciences 177(1), 28–40 (2007)
14. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177(1), 41–73 (2007)
15. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. Wiley (2008)
16. Pedrycz, W., Vukovich, G.: Granular computing with shadowed sets. International Journal of Intelligent Systems 17, 173–197 (2002)
17. Pedrycz, W.: Shadowed sets: Representing and processing fuzzy sets. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 28(1), 103–109 (1998)
18. Peters, J.: Approximation and perception in ethology-based reinforcement learning. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Granular Computing, pp. 671–688. John Wiley & Sons, Ltd., Chichester (2008)
19. Peters, J.: Near Sets. special Theory about Nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
20. Priss, U.: Formal Concept Analysis in Information Science. In: Cronin, B. (ed.) Annual Review of Information Science and Technology, vol. 40, pp. 521–543. Information Today, Inc. on behalf of ASIS&T, Medford (2006)
21. Skowron, A., Peters, J.: Rough-granular computing. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Granular Computing, pp. 225–328. John Wiley & Sons, Ltd., Chichester (2008)
22. Tsumoto, S., Tanaka, H.: Discovery of Approximate Medical Knowledge Based on Rough Set Model. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 468–476. Springer, Heidelberg (1998)
23. Yao, Y.Y.: Granular computing using neighborhood systems. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) Advances in Soft Computing: Engineering Design and Manufacturing, The 3rd On-line World Conference on Soft Computing, WSC3, June 21-30, pp. 539–553. Springer, London (1999)

24. Yao, Y.Y.: Information Granulation and Rough Set Approximation. International Journal of Information Systems 16, 87–104 (2001)
25. Yao, Y. Y., Yao, J.T.: Granular Computing as a Basis for Consistent Classification Problems. In: Proceedings of PAKDD 2002 Workshop on Toward the Foundation of Data Mining, pp. 101–106 (2002)
26. Yao, Y.Y.: Mining high order decision rules. In: Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.) Rough Set Theory and Granular Computing, pp. 125–135. Springer, Berlin (2003)
27. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Chapter 23
Rough Sets and Medical Differential Diagnosis

Shusaku Tsumoto

Abstract. This chapter discusses a correspondence between the core ideas of rough sets and medical differential diagnosis. Classically, a disease is defined as a set of symptoms, each of which gives the degree of confidence and coverage for the diagnosis. Diagnostic procedure mainly consists of the following three procedures: First, focusing mechanism (characterization) selects the candidates of differential diagnosis by using a set of symptoms. Secondly, additional set of symptoms make a differential diagnosis among the selected candidates. Finally, complications of other disease will be considered by symptoms which cannot be explained by the final candidates. This chapter mainly focuses on the first and second process and shows that these processes correspond to rules extracted by upper and lower approximation of supporting set of a given disease.

Keywords: Rough sets, data mining, rule induction, focusing mechanism, medical diagnostic model.

23.1 Introduction

Due to the rapid progress of biomolecular technologies, medical diagnostic, and therapeutic process are chaging very rapidly. In other words, laboratory examinations, radiological examinations, and molecular- based drug development are indispenable for clinical activities. However, the etiology of many diseases are still unknown, whose diagnosis should be made by classical methods, such as interviewing patient history and physical examinations.

Classical medical diagnosis of a disease assumes that a disease is defined as a set of symptoms. If symptoms are observed enough, a set of symptoms give some

Shusaku Tsumoto
Department of Medical Informatics, School of Medicine, Faculty of Medicine,
Shimane University
e-mail: tsumoto@med.shimane-u.ac.jp

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 605–621.
springerlink.com

confidence to diagnosis of a corresponding disease. Thus, correspondence between a set of manifestations and a disease will be used to make its diagnosis. Such sets of symptoms may be used for differential diagnosis, and similarity of diseases will be infered by sets of symptoms.

This chapter shows that such a diagnostic process can be modeled by the core ideas of rough sets: selection of candidates (screening) and differential diagnosis are closely related with diagnostic rules obtained by upper and lower approximations of a given concept.

The chapter is organized as follows. Section 23.2 shows characteristics of medical diagnostic process. Section 23.3 discusses correspondence between medical diagnosis and the core ideas of rough sets. Section 23.4 provides algorithms for induction of diagnostic rules. Section 23.5 gives an extention of the above ideas in probabilistic domain, which can be viewed as application of variable precision rough set model [19]. Section 23.6 discussed what has not been achieved yet. Finally, Section 23.7 concludes this chapter

23.2 Medical Diagnostic Process

23.2.1 Differential Diagnosis of Headache

Headache is one of the most important chief complaints in Neurological Diseases. Although most of the etiology of headache is unknown and classified into *Primary Headache*, headache with other disorders, called *Secondary Headache* may need emergent care. In some cases, neuroimaging or other laboratory/physiological examinations cannot capture the need for emergency action, classical empirical examinations are still very important, compared with other medical fields The crucial point of differential diagnosis is whether its focus is intracranial or not: if the disease is caused by intracranial disorders, intensive care should be taken immediately. However, most of cases of headache (70%) are coming from muscle pain, which may not need emergency action. Thus, differential diagnosis is very important

23.2.2 Classification of Headache

Due to the importance of classification and diagnosis of headache, International Headache Society published the second version of *International Classification of Headache* (ICHD-II) in 2004 [1]. This classification gives the definition and diagnostic criteria of a disease. The basis of this classification is the following:

1. Single classification for all purposes
2. Comprehensive
3. Evidence-based as far as possible

4. Symptom-based for the primary headaches, aetiological for the secondary headaches
5. Unambiguous terms such as sometimes, often, usually are avoided
6. Specificity weighted over sensitivity
7. Separate codes for probable cases

The classification system is hierarchical and phenomenological: the major groups consist of three types: (1) primary headache, (2) secondary headache and (3) cranial neuralgias, central and primary facial pain and other headaches. Primary headache disorders are headache without any causative disorder, and consists of the following four types:

1. Migraine
2. Tension-type headache
3. Cluster headache and other trigeminal autonomic cephalalgias
4. Other primary headaches

The above classification can be summarized into the following interesting characteristics:

(a) Hierarchical classification is used.
(b) A set of Symptoms is used to describe each disease.
(c) Description is based on specificity weighted over sensitivity, which shows that reasoning about frequency is implicitly included.
(d) For coverage, exceptions are described.
(e) Diagnostic criteria give temporal information about episodes of headache.

Since the author was involved in developing an expert system for differential diagnosis of headache [2], he focused on the nature of description and frequency as shown in Section 23.3. The informal history about how the author reached rough sets is shown in Appendix.

23.2.3 Examples: Migraine and Tension-Type Headache

For example, Figures 23.1 and 23.2 show the cases of migraine without aura, formally called common migraine and infrequent episodic tension-type headache.

These examples give us several interesting characteristics for differential diagnosis: the diagnostic criterion [D.] of tension-type headache is negation of that of migraine. The criteria [C.] show different symptoms, which are useful for differential diagnosis. Thus, [D.] can be applied to negation of one of the diseases, and [C.] can be applied to confirmation of one of the diseases. Therefore, such combination of negation and confirmation is very important for diagnosis of headache, which we call **focusing mechanism**, discussed later.

1.1 Migraine without aura G43.0
Previously used terms common migraine, hemicrania simplex
Description: Recurrent headache disorder manifesting in attacks lasting 4-72 hours. Typical characteristics of the headache are unilateral location, pulsating quality, moderate, or severe intensity, aggravation by routine physical activity and association with nausea and/or photophobia and phonophobia
Diagnostic criteria:

A. At least 5 attacks_1 fulfilling criteria B-D
B. Headache attacks lasting 4-72 hours (untreated or unsuccessfully treated)
C. Headache has at least two of the following characteristics:
    a. unilateral location
    b. pulsating quality
    c. moderate or severe pain intensity
    d. aggravation by or causing avoidance of routine physical activity (eg, walking or climbing stairs)
D. During headache at least one of the following:
    a. nausea and/or vomiting
    b. photophobia and phonophobia
E. Not attributed to another disorder

Fig. 23.1. Migraine without aura[1]

23.2.4 Focusing Mechanism

One of the characteristics in medical reasoning is a focusing mechanism, which is used to select the final diagnosis from many candidates [18], [10]. For example, in differential diagnosis of headache, more than 60 diseases will be checked by present history, physical examinations, and laboratory examinations. In diagnostic procedures, a candidate is excluded if a symptom necessary to diagnose is not observed.

This style of reasoning consists of the following two processes: exclusive reasoning and inclusive reasoning. Relations of this diagnostic model with another diagnostic model are discussed in [8]. The diagnostic procedure proceeds as follows (Fig. 23.3): First, exclusive reasoning excludes a disease from candidates when a patient does not have a symptom that is necessary to diagnose that disease. Second, inclusive reasoning suspects a disease in the output of the exclusive process when a patient has symptoms specific to a disease. These two steps are modeled as two kinds of rules, negative rules (or exclusive rules) and positive rules; the former corresponds to exclusive reasoning, the latter to inclusive reasoning. In the next two sections, these two rules are represented as deterministic rules, which can be viewed as special kinds of probabilistic rules.

2.1 Infrequent episodic tension-type headache G44.2
Description: Infrequent episodes of headache lasting minutes to days. The pain is typically bilateral, pressing or tightening in quality and of mild-to-moderate intensity, and it does not worsen with routine physical activity. There is no nausea but photophobia or phonophobia may be present.
Diagnostic criteria:

A. At least 10 episodes occurring on ¡1 day per month on average (¡12 days per year) and fulfilling criteria B-D
B. Headache lasting from 30 minutes to 7 days
C. Headache has at least two of the following characteristics:
   a. bilateral location
   b. pressing/tightening (non-pulsating) quality
   c. mild or moderate intensity
   d. not aggravated by routine physical activity such as walking or climbing stairs
D. Both of the following:
   a. no nausea or vomiting (anorexia may occur)
   b. no more than one of photophobia or phonophobia
E. Not attributed to another disorder

Fig. 23.2. Infrequent episodic tension-type headache[1]

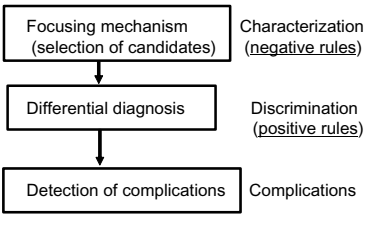


Fig. 23.3. Illustration of focusing mechanism

23.3 Definition of Rules

23.3.1 Rough Sets

In the following sections, we use the following notation introduced by Grzymała-Busse and Skowron [7], based on rough set theory [3]. Let U denote a nonempty finite set called the universe and A denote a nonempty, finite set of attributes, that is, $a : U \rightarrow V_a$ for $a \in A$, where V_a is called the domain of a, respectively. Then a decision table is defined as an information system, $A = (U, A \cup \{d\})$. The atomic formulas over $B \subseteq A \cup \{d\}$ and V are expressions of the form $[a = v]$, called descriptors over B, where $a \in B$ and $v \in V_a$. The set $F(B,V)$ of formulas over B is the least set containing all atomic formulas over B and closed with respect to disjunction, conjunction, and negation.

For each $f \in F(B,V)$, f_A denotes the meaning of f in A, that is, the set of all objects in U with property f, defined inductively as follows:

1. If f is of the form $[a = v]$, then $f_A = \{s \in U | a(s) = v\}$.
2. $(f \wedge g)_A = f_A \cap g_A$; $(f \vee g)_A = f_A \vee g_A$; $(\neg f)_A = U - f_a$.

23.3.2 Classification Accuracy and Coverage

23.3.2.1 Definition of Accuracy and Coverage

By use of the preceding framework, classification accuracy and coverage, or true positive rate are defined as follows.

Definition 23.1. Let R and D denote a formula in $F(B,V)$ and a set of objects that belong to a decision d. Classification accuracy and coverage(true positive rate) for $R \rightarrow d$ is defined as:

$$\alpha_R(D) = \frac{|R_A \cap D|}{|R_A|} (= P(D|R)), \tag{23.1}$$

$$\kappa_R(D) = \frac{|R_A \cap D|}{|D|} (= P(R|D)), \tag{23.2}$$

where $|S|$, $\alpha_R(D)$, $\kappa_R(D)$, and $P(S)$ denote the cardinality of a set S, a classification accuracy of R as to classification of D, and coverage (a true positive rate of R to D), and probability of S, respectively.

Figure 23.4 depicts the Venn diagram of relations between accuracy and coverage. Accuracy views the overlapped region $|R_A \cap D|$ from the meaning of a relation R. On the other hand, coverage views the overlapped region from the meaning of a concept D.

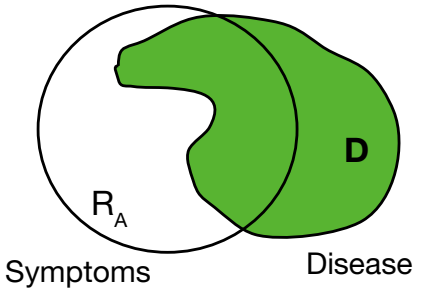


Fig. 23.4. Venn diagram of accuracy and coverage

It is notable that $\alpha_R(D)$ measures the degree of the sufficiency of a proposition, $R \rightarrow D$, and that $\kappa_R(D)$ measures the degree of its necessity. For example, if $\alpha_R(D)$ is equal to 1.0, then $R \rightarrow D$ is true. On the other hand, if $\kappa_R(D)$ is equal to 1.0, then $D \rightarrow R$ is true. Thus, if both measures are 1.0, then $R \leftrightarrow D$.

23.3.3 Probabilistic Rules

By use of accuracy and coverage, a probabilistic rule is defined as:

$$R \overset{\alpha,\kappa}{\rightarrow} d \quad s.t. \quad R = \wedge_j [a_j = v_k], \alpha_R(D) \geq \delta_\alpha \text{ and } \kappa_R(D) \geq \delta_\kappa. \tag{23.3}$$

If the thresholds for accuracy and coverage are set to high values, the meaning of the conditional part of probabilistic rules corresponds to the highly overlapped region. Figure 23.5 depicts the Venn diagram of probabilistic rules with highly overlapped region. This rule is a kind of probabilistic proposition with two statistical measures, which is an extension of Ziarko's variable precision model (VPRS) [19][1].

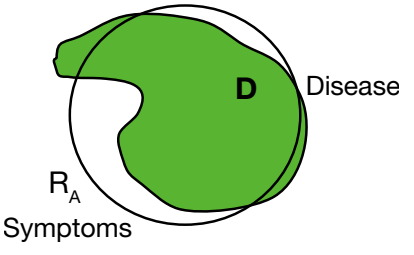


Fig. 23.5. Venn diagram for probabilistic rules

It is also notable that both a positive rule and a negative rule are defined as special cases of this rule, as shown in the next sections.

23.3.4 Positive Rules

A positive rule is defined as a rule supported by only positive examples, the classification accuracy of which is equal to 1.0. It is notable that the set supporting this rule corresponds to a subset of the lower approximation of a target concept, which is introduced in rough sets [3]. Thus, a positive rule is represented as:

$$R \rightarrow d \quad s.t. \qquad R = \wedge_j [a_j = v_k], \quad \alpha_R(D) = 1.0.$$

[1] This probabilistic rule is also a kind of *rough modus ponens* [5].

Figure 23.6 shows the Venn diagram of a positive rule. As shown in this figure, the meaning of R is a subset of that of D. This diagram is exactly equivalent to the classic proposition $R \rightarrow d$.

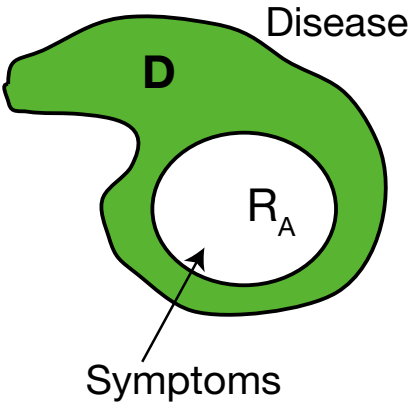


Fig. 23.6. Venn diagram of positive rules

This positive rule is often called a deterministic rule. However, we use the term, positive (deterministic) rules, because a deterministic rule supported only by negative examples, called a negative rule, is introduced in the next section.

23.3.5 Negative Rules

Before defining a negative rule, let us first introduce an exclusive rule, the contrapositive of a negative rule [18]. An exclusive rule is defined as a rule supported by all the positive examples, the coverage of which is equal to 1.0. That is, an exclusive rule represents the necessity condition of a decision. It is notable that the set supporting an exclusive rule corresponds to the upper approximation of a target concept, which is introduced in rough sets [3]. Thus, an exclusive rule is represented as:

$$R \rightarrow d \quad s.t. \qquad R = \vee_j [a_j = v_k], \quad \kappa_R(D) = 1.0.$$

Figure 23.7 shows the Venn diagram of an exclusive rule. As shown in this figure, the meaning of R is a superset of that of D. This diagram is exactly equivalent to the classic proposition $d \rightarrow R$.

From the viewpoint of propositional logic, an exclusive rule should be represented as:

$$d \rightarrow \vee_j [a_j = v_k],$$

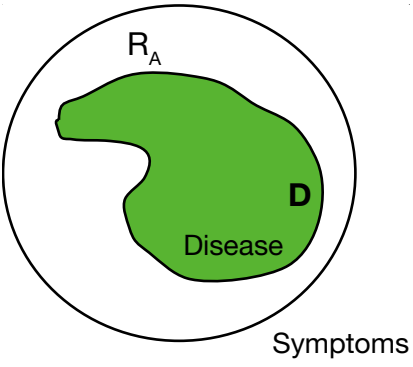


Fig. 23.7. Venn diagram of exclusive rules

because the condition of an exclusive rule corresponds to the necessity condition of conclusion d. Thus, it is easy to see that a negative rule is defined as the contrapositive of an exclusive rule:

$$\wedge_j \neg[a_j = v_k] \rightarrow \neg d,$$

which means that if a case does not satisfy any attribute value pairs in the condition of a negative rule, then we can exclude a decision d from candidates.

In summary, a negative rule is defined as:

$$\wedge_j \neg[a_j = v_k] \rightarrow \neg d \quad s.t. \quad \forall[a_j = v_k]\ \kappa_{[a_j = v_k]}(D) = 1.0,$$

where D denotes a set of samples that belong to a class d. Figure 23.8 shows the Venn diagram of a negative rule. As shown in this figure, it is notable that this negative region is the "positive region" of "negative concept."

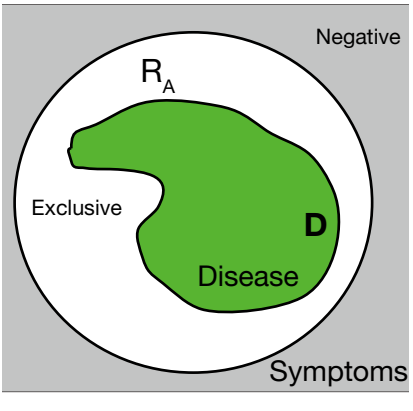


Fig. 23.8. Venn diagram of negative rules

Negative rules should also be included in a category of deterministic rules, because their coverage, a measure of negative concepts, is equal to 1.0. It is also notable that the set supporting a negative rule corresponds to a subset of negative region, which is introduced in rough sets [3].

In summary, positive and negative rules correspond to positive and negative regions defined in rough sets. Figure 23.9 shows the Venn diagram of those rules.

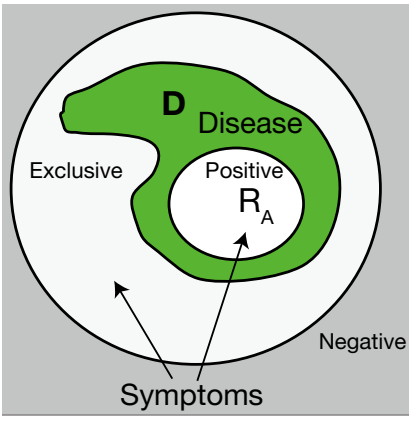


Fig. 23.9. Venn diagram of defined rules

23.4 Algorithms for Rule Induction

The contrapositive of a negative rule, an exclusive rule, is induced as an exclusive rule by the modification of the algorithm introduced in PRIMEROSE-REX [18, 8], as shown in Fig. 23.10 [12, 11]. This algorithm works as follows. (1) First it selects a descriptor $[a_i = v_j]$ from the list of attribute-value pairs, denoted by L. (2) Then it checks whether this descriptor overlaps with a set of positive examples, denoted by D. (3) If so, this descriptor is included in a list of candidates for positive rules and the algorithm checks whether its coverage is equal to 1.0. If the coverage is equal to 1.0, then this descriptor is added to $R_e r$, the formula for the conditional part of the exclusive rule of D. (4) Then $[a_i = v_j]$ is deleted from the list L. This procedure, from (1) to (4), will continue unless L is empty. (5) Finally, when L is empty, this algorithm generates negative rules by taking the contrapositive of induced exclusive rules.

On the other hand, positive rules are induced as inclusive rules by the algorithm introduced in PRIMEROSE-REX [18], as shown in Fig. 23.11. For induction of positive rules, the threshold of accuracy and coverage is set to 1.0 and 0.0, respectively.

```
procedure Exclusive and Negative Rules;
  var
    L : List;
      /* A list of elementary attribute-value pairs */
  begin
    L := P0;
    /* P0: A list of elementary attribute-value pairs given in a database */
    while (L ≠ {}) do
      begin
        Select one pair [ai = vj] from L;
        if ([ai = vj]A ∩ D ≠ ϕ) then do /* D: positive examples of a target class d */
          begin
          Lir := Lir + [ai = vj];    /* Candidates for Positive Rules */
          if (κ[ai=vj](D) = 1.0)
          then Rer := Rer ∧ [ai = vj];
            /* Include [ai = vj] into the formula of Exclusive Rule */
          end
      L := L − [ai = vj];
      end
    Construct Negative Rules:
        Take the contrapositive of Rer.
  end {Exclusive and Negative Rules};
```

Fig. 23.10. Induction of exclusive and negative rules

This algorithm works in the following way. (1) First it substitutes L_1, which denotes a list of formulas composed of only one descriptor, with the list L_{er} generated by the former algorithm shown in Fig. 9.1. (2) Then, until L_1 becomes empty, the following steps will continue: (a) A formula $[a_i = v_j]$ is removed from L_1. (b) Then, the algorithm checks whether $\alpha_R(D)$ is larger than the threshold. (For induction of positive rules, this is equal to checking whether $\alpha_R(D)$ is equal to 1.0.) If so, then this formula is included a list of the conditional parts of positive rules. Otherwise, it will be included in M, which is used for making conjunctions. (3) When L_1 is empty, the next list L_2 is generated from the list M.

23.5 Extension into Variable Precision Rough Set Model

Although the above diagnostic model clearly shows the coresspondence to the original rough set model, unfortunately, medical domain is not deterministic. High specificity is preferable, but it is very difficult to obtain the full specificity without loss of predictability. This is why RHINOS focuses on inclusive rules, which can be viewed as an probabilistic extention of positive rule. Interestingly, this extension corresponds to lower approximation of variable precision rough set model [19]. The definition of inclusive rules can be obtained as Equation 23.3 with high accuracy and

```
procedure Positive Rules;
  var
    i : integer;    M, L_i : List;
  begin
    L_1 := L_ir;
    /* L_ir: A list of candidates generated by induction of exclusive rules */
    i := 1;    M := {};
    for i := 1 to n do
    /* n: Total number of attributes given
       in a database */
      begin
        while ( L_i ≠ {} ) do
          begin
            Select one pair R = ∧[a_i = v_j] from L_i;
            L_i := L_i − {R};
            if    (α_R(D) > δ_α)
              then  do S_ir := S_ir + {R};
          /* Include R in a list of the Positive Rules */
            else M := M + {R};
          end
        L_{i+1} := (A list of the whole combination of the conjunction formulae in M);
      end
  end {Positive Rules};
```

Fig. 23.11. Induction of positive rules

high coverage. Thus, an algorithm for induction of inclusive rules can be achieved by small modification of Fig 23.11, as shown in Fig 23.12.

The author conducted several experimental evaluations on several neurological datasets [8, 12, 11], which shows that the performance of these algorithms are better than conventional methods.

23.6 Discussion: What Has Not Been Achieved?

In Section 23.2, the author discusses the characteristics of classification of headache: (a) hierarchical classification is used. (b) A set of Symptoms is used to describe each disease. (c) Description is based on specificity weighted over sensitivity, which shows that reasoning about frequency is implicitly included. (d) For coverage, exceptions are described. (e) Diagnostic criteria give temporal information about episodes of headache.

In the above studies, automated extraction of knowledge with respect to (b), (c), and (d) has been solved. Concerning (a), the author proposed several approach to mining taxonomy from a dataset in [9, 13, 14].

However, acquisition of temporal knowledge has not been achieved yet. When the author interviewed the domain expert for RHINOS, he found that temporal

```
procedure Positive Rules;
  var
    i : integer;    M, L_i : List;
  begin
    L_1 := L_ir;
    /* L_ir: A list of candidates generated by induction of exclusive rules */
    i := 1;    M := {};
    for i := 1 to n do
    /* n: Total number of attributes given
       in a database */
      begin
        while ( L_i ≠ {} ) do
          begin
            Select one pair R = ∧[a_i = v_j] from L_i;
            L_i := L_i − {R};
            if   (α_R(D) > δ_α) and (κ_R(D) > δ_κ)
              then do S_ir := S_ir + {R};
          /* Include R in a list of the Inclusive Rules */
            else M := M + {R};
          end
        L_{i+1} := (A list of the whole combination of the conjunction formulae in M);
      end
  end {Positive Rules};
```

Fig. 23.12. Induction of positive rules

reasoning is very important for complicated cases. For example, one patient suffers from both common migraine and tension headache. According to the diagnostic rules, RHINOS diagnoses the case as migraine. However, the main complaint came from tension headache. Since the onset of tension headache is persistent but the severity is mild, the patient focuses on the symptoms of migraine. If the system can focus on the differences in temporal natures of headaches, then it can detect the complications of migraine and tension headache. Thus, temporal reasoning is a key to diagnose completed cases especially when all the symptoms may give a contradict interpretation.

Research on temporal data mining is ongoing, and now the authors realized that temporal data mining is very important for risk management in several fields [15, 17, 16]. It will be our future work to develop methodologies for temporal rule mining in clinical data.

23.7 Conclusion

This paper focuses on medical diagnostic process and discusses correspondence between the process and rough set model. The key characteristics in medical reasoning

is a focusing mechanism: First, exclusive reasoning excludes a disease from candidates when a patient does not have a symptom that is necessary to diagnose that disease. Second, inclusive reasoning suspects a disease in the output of the exclusive process when a patient has symptoms specific to a disease. These two steps are modeled as two kinds of rules obtained from representations of upper and lower approximation of a given disease. Thus, rule induction based on rough set model is a powerful tool for automated extraction or mining of rules following a focusing mechanism from dataset.

Acknowledgements. The author would like to thank late Professor Pawlak for all the comments on the author's research and his encouragement. Without his influence, the author would neither have received PhD on computer science nor become a professor of medical informatics. The author also would like to thank Professor Jerzy Grzymała-Busse, Andrzej Skowron, Roman Słowiński, and Wojciech Ziarko for their insightful comments.

Appendix: RHINOS

RHINOS is an expert system which diagnoses clinical cases on headache or facial pain from manifestations. In this system, a diagnostic model proposed by Matsumura
[2] is applied to the domain, which consists of the following three kinds of reasoning processes: exclusive reasoning, inclusive reasoning, and reasoning about complications.

First, exclusive reasoning excludes a disease from candidates when a patient does not have a symptom which is necessary to diagnose that disease. Secondly, inclusive reasoning suspects a disease in the output of the exclusive process when a patient has symptoms specific to a disease. Finally, reasoning about complications suspects complications of other diseases when some symptoms which cannot be explained by the diagnostic conclusion are obtained.

Each reasoning is rule-based, and all the rules needed for diagnostic processes are acquired from medical experts in the following way.

(1) Exclusive Rules

These rule correspond to exclusive reasoning. In other words, the premise of this rule is equivalent to the necessity condition of a diagnostic conclusion. From the discussion with medical experts, the following six basic attributes are selected which are minimally indispensable for defining the necessity condition: *1. Age, 2. Pain location, 3. Nature of the pain, 4. Severity of the pain, 5. History since onset,*

6. Existence of jolt headache. For example, the exclusive rule of common migraine is defined as:

```
In order to suspect common migraine,
the following symptoms are required:
pain location: not eyes,
nature :throbbing or persistent or radiating,
history: paroxysmal or sudden and
jolt headache: positive.
```

One of the reasons why the six attributes are selected is to solve an interface problem of expert systems: if all attributes are considered, all the symptoms should be input, including symptoms which are not needed for diagnosis. To make exclusive reasoning compact, we chose the minimal requirements only. It is notable that this kind of selection can be viewed as the ordering of given attributes, which is expected to be induced from databases. This issue is discussed later in Section 6.

(2) Inclusive Rules

The premises of inclusive rules are composed of a set of manifestations specific to a disease to be included. If a patient satisfies one set, this disease should be suspected with some probability. This rule is derived by asking the medical experts about the following items for each disease: *1. a set of manifestations by which we strongly suspect a disease. 2. the probability that a patient has the disease with this set of manifestations:SI(Satisfactory Index) 3. the ratio of the patients who satisfy the set to all the patients of this disease: CI(Covering Index) 4. If the total sum of the derived CI(tCI) is equal to 1.0 then end. Otherwise, goto 5. 5. For the patients with this disease who do not satisfy all the collected set of manifestations, goto 1.* Therefore, a positive rule is described by a set of manifestations, its satisfactory index (SI), which corresponds to *accuracy measure*, and its covering index (CI), which corresponds to *total positive rate*. Note that, SI and CI are given empirically by medical experts.

For example, one of three positive rules for common migraine is given as follows.

```
If history: paroxysmal, jolt headache: yes,
nature: throbbing or persistent,
prodrome: no, intermittent symptom: no,
persistent time: more than 6 hours,
and location: not eye,
then common migraine is suspected with
accuracy 0.9 (SI=0.9) and this rule covers
60 percent of the total cases (CI=0.6).
```

(3) Disease Image

This rule is used to detect complications of multiple diseases, acquired by all the possible manifestations of the disease. By the use of this rule, the manifestations that cannot be explained by the conclusions will be checked, which suggest complications of other diseases. For example, the disease image of common migraine is as follows:

```
The following symptoms can be explained by
common migraine: pain location: any or
depressing: not or jolt headache: yes or ...
```

Therefore, when a patient who suffers from common migraine is depressing, it is suspected that he or she may also have other diseases.

References

1. Headache Classification Subcommittee of the International Headache Society: The international classification of headache disorders. Cephalalgia, 2nd edn., vol. 24 (suppl. 1), pp. 9–160 (2004)
2. Matsumura, Y., Matsunaga, T., Maeda, Y., Tsumoto, S., Matsumura, H., Kimura, M.: Consultation System for Diagnosis of Headache and Facial Pain: "Rhinos". In: Wada, E. (ed.) LP 1985. LNCS, vol. 221, pp. 287–298. Springer, Heidelberg (1986)
3. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers, Dordrecht (1991)
4. Pawlak, Z.: Anatomy of conflics. Bulletin of the EATCS 50, 234–246 (1993)
5. Pawlak, Z.: Rough modus ponens. In: Proc. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 1998, pp. 1162–1166. La Sorbone, Paris (1998)
6. Pawlak, Z.: Some remarks on conflict analysis. European Journal of Operational Research 166(3), 649–654 (2005)
7. Skowron, A., Grzymała-Busse, J.: From rough set theory to evidence theory. In: Yager, R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 193–236. John Wiley & Sons, New York (1994)
8. Tsumoto, S.: Automated induction of medical expert system rules from clinical databases based on rough set theory. Information Sciences 112, 67–84 (1998)
9. Tsumoto, S.: Extraction of experts' decision rules from clinical databases using rough set model. Intelligent Data Analysis 2(3), 215–227 (1998)
10. Tsumoto, S.: Modelling Medical Diagnostic Rules Based on Rough Sets. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 475–482. Springer, Heidelberg (1998)
11. Tsumoto, S.: Automated discovery of positive and negative knowledge in clinical databases based on rough set model. IEEE Engineering in Medicine and Biology Magazine 19, 56–62 (2000)
12. Tsumoto, S.: Knowledge discovery in clinical databases and evaluation of discovered knowledge in outpatient clinic. Information Sciences 124, 125–137 (2000)
13. Tsumoto, S.: Automated extraction of hierarchical decision rules from clinical databases using rough sets. Expert Systems with Applications 24, 189–197 (2003)

14. Tsumoto, S.: Extraction of structure of medical diagnosis from clinical data. Fundam. Inform. 59(2-3), 271–285 (2004)
15. Tsumoto, S., Hirano, S.: Risk mining in medicine: Application of data mining to medical risk management. Fundam. Inform. 98(1), 107–121 (2010)
16. Tsumoto, S., Hirano, S.: Detection of risk factors using trajectory mining. J. Intell. Inf. Syst. 36(3), 403–425 (2011)
17. Tsumoto, S., Hong, T.P.: Special issue on data mining for decision making and risk management. J. Intell. Inf. Syst. 36(3), 249–251 (2011)
18. Tsumoto, S., Tanaka, H.: Automated discovery of medical expert system rules from clinical databases based on rough sets. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining 1996, pp. 63–69. AAAI Press, Palo Alto (1996)
19. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)

Chapter 24
Science and Semantics: A Note on Rough Sets and Vagueness

Marcin Wolski

Abstract. In the chapter we present rough set theory against the background of recent philosophical discussions about vagueness and empirical sciences. Weiner, in her article about this topic, discusses the supervaluationist semantics of vague predicates and its criticism offered by Fodor and Lepore. She argues that neither the former nor latter approach is consistent with the scientific methodology of dealing with vague concepts such as "obese". In actual fact, it is Frege's philosophical approach that concepts must have sharp boundaries, which is the closest to scientific practice. In this context, rough set theory can be viewed as a modified supervaluationist semantics. To be more precise, rough sets provide a modal version of this semantics, where the super-truth is replaced by a local one. However, there are flies in the ointment: firstly, rough set theory is philosophically weaker than supervaluationism (in consequence, more vulnerable to the criticism of Fodor and Lepore); secondly, Weiner's arguments concerning scientific methods apply to rough sets as well. Yet there is also good news: this philosophical weakness stays actually in full accordance with scientific practice. Thus, rough set theory may be seen as a supervaluationism shifted toward the scientific methodology. In the chapter we shall make a further step into this direction and also present how rough set theory would be like when made fully consistent with the scientific approach to vague predicates. In other words, we also offer a Fregean rough set methodology.

Keywords: Vagueness, supervaluationism, rough sets, methodology of empirical sciences, philosophy of language.

Marcin Wolski
Department of Logic and Philosophy of Science
Maria Curie-Skłodowska University, Poland
e-mail: marcin.wolski@umcs.lublin.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 623–643.
springerlink.com

24.1 Introduction

So far, rough set theory and the theories of vagueness have been discussed together mainly in connection to fuzzy sets, see *e.g.*, [1, 15, 16]. As is well known, the theory of fuzzy sets belongs to the category of degree theories of vagueness or, to use a different term, many-valued logics approach. This theory is widely recognized as very important in computer science; yet, in philosophy, it is much less popular; as Kefee put it in her book [6]: *[b]ut Zadeh will not figure centrally in this chapter since his philosophical contribution is less significant*. In this chapter, we would like to connect rough sets with some other theories of vagueness which are also philosophically important. To this end, we shall present recent philosophical discussions concerned with vagueness as it is dealt with in both philosophy and empirical sciences.

Weiner in her excellent article [19] observes that the methods and results of science have not yet been taken into account in the philosophical discussions of the semantics for vague predicates. Worse still, it is widely assumed that in these discussions no empirical data need to be considered seriously. In the Stanford Encyclopedia of Philosophy in the entry "Vagueness" one can find:

Vagueness is standardly defined as the possession of borderline cases. For example, 'tall' is vague because a man who is 1.8 meters in height is neither clearly tall nor clearly non-tall. No amount of conceptual analysis or empirical investigation can settle whether a 1.8 meter man is tall. Borderline cases are inquiry resistant. Indeed, the inquiry resistance typically recurses. For in addition to the unclarity of the borderline case, there is normally unclarity as to where the unclarity begins. In other words 'borderline case' has borderline cases. This higher-order vagueness shows that 'vague' is vague.

Thus, vagueness is defined as the possession of borderline cases which are inquiry resistant. Furthermore, vagueness is always higher-order (*'borderline case' has borderline cases*). However, as Weiner argues, this view stays in conflict with the scientific methodology. She observes that such vague predicates as "obese" play also a central role in science, *for example* in medical research. On the National Institute of Health Obesity Research web page one can find:

Obesity is a major contributor to serious health conditions in children and adults, including type 2 diabetes, cardiovascular disease, many forms of cancer, and numerous other diseases and conditions.

Thus, this vague concept has already been scientifically examined. Better still, in order to compute statistical correlations between obesity and cardiovascular disease, we have to understand what the vague term "obesity" *exactly* (sic!) means in medical sciences. It turns out that vague predicates which might be used in scientific empirical investigations must be tied to the process of precisification, that is, the process of ruling out borderline cases. Thus, scientific practice is opposite to the basic philosophical assumptions concerning vagueness.

Of course, the above philosophical assumptions are shared by computer scientists as well. As Pawlak observes [12]:

For example, the concept of an 'odd' ('even') number is crisp, whereas the concept of a 'beautiful woman' is vague because for some women we cannot decide whether they are beautiful or not (they are boundary-line cases).

In the rough set approach vagueness is due to lack of information about some elements of the universe. If with some elements the same information is associated, in view of this information these elements are indiscernible. [...] It turns out that indiscernibility leads to the boundary-line cases, i.e. some elements cannot be classified to the concept or its complement in view of the information available and thus form boundary-line cases.

The main aim of this article is to discuss the rough set methodology against the background of recent philosophical discussions concerning vagueness and empirical sciences. As said above, vague concepts in empirical sciences are defined by means a process of precisification. Since the very similar idea is also present in supervaluationism [2, 6], Weiner [19] starts her research with the supervaluationist semantics for vague predicates and its criticism offered by Fodor and Lepore [3]. Then she argues that neither former nor latter approach can be accepted without abandoning the scientific methodology. Finally, it turns out that an approach offered by Frege [5, 4] – for whom a vague predicate is not actually a predicate – is a philosophical approach expressing the way in which empirical sciences deal with vague concepts.[1]

Writing about rough sets in the above context needs a few words of caution. Rough set theory is a successful field of research, what has resulted in the divers extensions of this theory on other fields, including also the phenomenon of vagueness, *for example* [1, 16]. In consequence, the framework of rough set is very broad and one can connect it with a number of different approaches. In this chapter, we focus our attention on (basic) rough set theory as it was originally designed by Pawlak in his early papers; that is, a given decision table is fixed and the main goal of rough set analysis is to approximate a decision attribute. In our opinion, it is the best way to extract basic methodological and philosophical assumptions staying behind rough sets. Thus, in what follows when we write about rough set theory we mean basic rough set theory.

In the above context, rough set theory – as we shall show – is very closely related to the supervaluationist semantics [2, 6]. In spite of that, there is also a major difference: the supervaluationist semantics assumes the super-truth (which is global); in the case of rough sets, the super-truth is replaced by a local-truth, what results in S5 modal logic. Furthermore, rough set theory starts with information systems (i.e. concrete structures) instead of classical models (abstract structures). This difference touches the problem of higher-order vagueness, which in both theories is not obvious. Kefee argues that in the case of supervaluationism, there is a place for this type of vagueness mainly due to vagueness in meta-theory [6]; however, the issue is still disputable. Our standpoint on this issue agrees with Kefee's view, mainly due to usage of classical models. In the case of the basic rough sets, where the set of conditional attributes

[1] As is well known, in philosophy the concept of "concept" is very problematic and there are many definitions in use. However, in empirical sciences what a "concept" really is does not play any role. Therefore we put aside all philosophical discussions concerning the nature of a concept itself.

from an information system is fixed, one cannot model this type of vagueness. In order to do so, authors must extend the basic model and consider different sets of attributes or allow for change of data; in this way, the boundary region ceases to be crisp, *for example* [16]. As said above, in this chapter, we want to deal with basic rough set theory; therefore, we assume that rough set theory does not actually model higher-order vagueness. The lack of higher-order vagueness is – in our opinion – an advantage: there is no need to satisfy all philosophical requirements concerned with the concept of vagueness, especially when they stay in a conflict with good scientific practice.

Summing up, the basic treatment of vague predicates in both rough set theory and supervaluationism is essentially the same. This type of representation results in, as demonstrated by Weiner, a conflict with good scientific practice. On the other hand, the lack of higher-order vagueness stays in accordance with the scientific methodology. Therefore, in the article, we make further steps in this direction and show how rough set theory would be like when made fully consistent with scientific practice. Thus, given that Frege's view on vagueness is actually the scientific view [18], we also offer a Fregean rough set theory.

24.2 Obese: A Case Study

In this section, we present some philosophical remarks which are relevant to the semantics of vague predicates as it is defined in philosophical studies and in scientific research. Fortunately, there is at least one paradigmatic vague predicate which has become a subject of both philosophy and science, that is,"obesity". The section is based on two papers by Weiner [18, 19] and one paper by Fodor and Lepore [3]. To make the material accessible, we omit abstract philosophical discussions and focus on the case study of "obesity", which is the first predicate given as an example by the Stanford Encyclopedia of Philosophy in the entry "Vagueness".

Let us look at this Encyclopedia once again.

For instance, a boy may count as a borderline case of 'obese' because people cannot tell whether he is obese just by looking at him. A curious mother could try to settle the matter by calculating her boy's body mass index. The formula is to divide his weight (in kilograms) by the square of his height (in meters). If the value exceeds 30, this test counts him as obese. The calculation will itself leave some borderline cases. The mother could then use a weight-for-height chart. These charts are not entirely decisive because they do not reflect the ratio of fat to muscle, whether the child has large bones, and so on. The boy will only count as an absolute borderline case of 'obese' if no possible method of inquiry could settle whether he is obese. When we reach this stage, we start to suspect that our uncertainty is due to the concept of obesity rather than to our limited means of testing for obesity.

As one can see, this description of the term "obese" takes into account also a medical context, that is, the body mass index (BMI). It is very interesting, since Weiner in her article [19] also takes this index to define obesity, but her conclusions

are opposite to the ones offered by the Encyclopedia (we shall return to this issue soon). The entry suggests also that even in medical sciences we must encounter borderline cases when dealing with vague predicates.

This approach is defended by Fodor and Lepore [3] in their criticism of supervaluationist theory of vagueness. According to supervaluationism, a vague predicate, such as "obese", is given no definite extension. Rather, there is a range of possible extensions. These various possible extensions correspond to different ways of making the predicate precise; *for example* "obese" may be given the following extensions (in the literature these extensions are called precisifications): (1) BMI greater than 30, (2) BMI greater than 27, (3) BMI greater than 24. The supervaluationist semantics assumes that truth conditions involve quantifying over all these ways of making language precise. Thus, Adam who has BMI = 26 will be settled in the borderline area (he is obese only under the interpretation 2), whereas if Bruno's BMI = 35, then "Bruno is obese" will be settled as true, because on all the ways of making "obese" precise (i.e., 1, 2, 3), Bruno is classified as obese. According to Fodor and Lepore, this idea of making vague predicate precise is, however, implausible [3]. If a sentence "Adam is obese" is indeterminate in one model, then it must be so in every model. They reason as follows. The point of precisification is actually bivalence; it excludes models at which this sentence is neither true nor false. For a sentence "Adam is obese", the supervaluationism requires models at which the sentence is true and models at which the sentence if false. But, Fodor and Lepore argue, *there can be no models of either kind* [3], because it is what actually vagueness means.

Summing up, from the standpoint of philosophy what is vague is, indeed conceptually necessary, vague and if "Adam is obese" *is unsettled in the actual world, then it is unsettled in every world* [3]. However, from the standpoint of empirical science, this issue looks quite different. At the National Cancer Institute web page one can find out that:

In 2002, about 41,000 new cases of cancer in the United States were estimated to be due to obesity. This means that about 3.2 percent of all new cancers are linked to obesity. A recent report estimated that, in the United States, 14 percent of deaths from cancer in men and 20 percent of deaths in women were due to overweight and obesity.

If obesity must have borderline cases, then the following question must arise: how these statistics are computed? According to Fodor and Lepore, in science we deal with a "new-obese" which is not in actual fact and English expression (as they put it: *you cannot precisify English* [3]). However, it means that we must abandon the widely accepted scientific methodology. Worse still, it also means that the above citation from the National Cancer Institute web page is not even written in English. Weiner gives the following answer to the question at issue [19]:

What is the researcher to do?

She does not simply hope for the best and decide to deal with borderline cases when they arise – perhaps failing to classify them at all. Each study subject must be classified, otherwise there will be no way to determine proportions. [...] The solution, in obesity research, is to use a measurable index of weight relative to height – typically, body mass index (BMI). And the researcher will decide, antecedent to

beginning the study, on a sharp cut-off (say, BMI $>$ 30) between the obese and non-obese. It is essential to such an investigation to draw a line.

Thus, even the supervaluationist semantics cannot be accepted by scientist, because in science, we need a single precisification. Of course, not whole philosophy is against scientific practice. As Weiner explains, Frege's standpoint on vague predicates is along the line of the scientific methodology [18]:

Frege says,

If something fails to display a sharp boundary, it cannot be recognized in logic as a concept [...] [1984, p. 133)

[...] there can be no doubt that Frege believed, for almost all his career, that concepts must have sharp boundaries [Weiner. 1990, pp. 92-97]. The above quotation about the need for sharp boundaries is from a paper published in 1891, but the same view appears in "Foundations", which was published in 1884. And in the second volume of "Basic Laws", which was printed in 1903, Frege says that something that appears to be a concept but does not have sharp boundaries is "wrongly termed a concept" [1952, p. 139] and is "an inadmissible sham concept" [1952, p. 145].

Of course, Frege's requirement that concepts must have sharp boundaries is recognized among many philosophers as very implausible (it is no accident that the Stanford Encyclopedia of Philosophy in the entry "Vagueness" does not even mention Frege and his philosophy). However, Weiner's point is that this requirement may be made plausible in some contexts and that, as above, it is no accident that these contexts can be found in science. As a matter of fact, in empirical studies if one wants to obtain valid inferences (conclusions), one must start with precise classifications. Therefore, Weiner concludes that [18]:

If, as Dummett argues, it is not enough to regard Frege as a philosopher of logic, then Frege should be regarded not as a philosopher of language, but as a philosopher of science.

Before we go further with the analysis of obesity, we shall introduce in the next section rough set theory [9, 10, 11]. In this way, the further analysis can be made within the conceptual framework of rough sets.

24.3 Rough Set Theory

In the present section, we briefly recall basic concepts from rough set theory which are relevant to our study [9, 10, 11].

Definition 24.1 (Information System). A quadruple $I = \langle U, Att, Val, f \rangle$ is called an *information system*, where:

- U is a non–empty finite set of objects;
- Att is a non–empty finite set of attributes;
- $Val = \bigcup_{A \in Att} Val_A$, where Val_A is the (non-empty) value–domain of the attribute A;

- $f : U \times Att \mapsto Val$ is a partial information function, such that for all $A \in Att$ and $a \in U$, when $f(a,A)$ is defined, then $f(a,A) \in Val_A$.

If f is a total function, that is, $f(a,A)$ is defined for all $a \in U$ and $A \in Att$, then the information system I is called *complete*; otherwise, it is called *incomplete*. The reader may also consult [8], where the concept of information system was elaborated.

When f is generalized to a function from $U \times Att$ to $\mathcal{P}(Val)$, where $\mathcal{P}(Val)$ is the powerset of Val, then the information system is *nondeterministic*. In what follows, we focus our attention on complete and deterministic systems.

If we distinguish in an information system two disjoint classes of attributes C and D, called condition and decision attributes, respectively, then the system will be called a *decision table*.

An information system I gives rise to an equivalence relation E, called an *indiscernibility relation*, defined as:

$$E = \{(a,b) : \forall A \in Att. \forall X \in Val\ (f(a,A) = X \Leftrightarrow f(b,A) = X)\}.$$

Customarily, E is often written as $IND(Att)$, the partition induced by the relation $IND(Att)$ is denoted by $U/IND(Att)$, and $[a]_{IND(Att)}$ denotes the equivalence class of $IND(Att)$ defined by $a \in U$. A simple generalization of $(U, IND(Att))$ is given by the concept of an approximation space:

Definition 24.2 (Approximation Space). A pair (U,E), where U is a non-empty set, and E is an equivalence relation on U, is called an *approximation space*. A subset $X \subseteq U$ is called *definable* if $X = \bigcup \mathcal{Y}$ for some $\mathcal{Y} \subseteq U/E$, where U/E is the family of equivalence classes of E (the quotient set of E).

Definition 24.3 (Approximation Operators). Let (U,E) be an approximation space. For every concept $X \subseteq U$, its E-lower and E-upper *approximations* are defined as follows, respectively:

$$\underline{X} = \{a \in U : [a]_E \subseteq X\},$$

$$\overline{X} = \{a \in U : [a]_E \cap X \neq \emptyset\}.$$

By the usual abuse of language and notation, the operator $\underline{\ \ } : \mathcal{P}(U) \to \mathcal{P}(U)$ sending X to $\underline{X}$ will be called the *lower approximation operator*, whereas the operator $\overline{\ \ } : \mathcal{P}(U) \to \mathcal{P}(U)$ sending X to $\overline{X}$ will be called the *upper approximation operator*.

Definition 24.4 (Increasing Representation of Rough Sets). For an approximation space (U,E) and $X \subseteq U$, a pair $(\underline{X}, \overline{X})$ is called an *increasing representation of* X.

Definition 24.5 (Disjoint Representation of Rough Sets). For an approximation space (U,E) and $X \subseteq U$, a pair $(\underline{X}, U \setminus \overline{X})$ is called a *disjoint representation of* X.

The set $U \setminus \overline{X}$ is often called an *exterior of* X and denoted by $Ext(X)$, whereas $Bnd(X) = \overline{X} \setminus \underline{X}$ is the boundary region of X. Of course, the choice of representation depends on a context of application. In the context of modal systems, the increasing

representation is more handy. On the other hand, in the context of abstract algebras the disjoint representation is more preferable.

The language for object description in an information system $\langle U, Att, Val, f\rangle$ is the descriptor language $\mathcal{L}_{Desc}$ [11]; its primitive formulas (atoms) are descriptors of the form $[A = val]$, where $A \in Att$ and $val \in Val_A$, which is read as *an attribute A has a value val*:

$$\alpha ::= [A = val] \mid \neg\alpha \mid \alpha \wedge \beta \mid \alpha \vee \beta$$

The set of atoms is denoted by Φ, and $\mathcal{F}_{Desc}$ denotes the set of all well-formed formulas.

Definition 24.6 ($\mathcal{L}_{Desc}$ Model). Let $\mathcal{L}_{Desc}$ be a descriptor language over a complete and deterministic information system $\rangle U, Att, Val, f\rangle$. Then $\rangle U, v\rangle$ is a model, where $v : U \times \Phi \to \{0,1\}$ is a function assigning to each pair (a, p), where $a \in U$ and $p \in \Phi$, a truth value. We usually write $v(a, p) = 1$ (or $v_a(p) = 1$), which is read as *for the object a, p is true*.

- $v_a([A = val]) = 1$ if $f(a,A) = val$, and 0 otherwise.

The function v is extended to every formula $\alpha \in \mathcal{F}_{Desc}$ in the standard way:

- $v_a(\neg\alpha) = 1$ if $v_a(\alpha) = 0$, and 0 otherwise,
- $v_a(\alpha \wedge \beta) = 1$ if $v_a(\alpha) = 1$ and $v_a(\beta) = 1$, and 0 otherwise.

Let be given a decision table, where C consists of n attributes and D consists of l attributes. Then each object $x \in U$ defines two formulas:

$$[A_1 = f(x,A_1)] \wedge [A_2 = f(x,A_2)] \ldots [A_n = f(x,A_n)],$$

where each $A_i \in C$. This formula is abbreviated as $C(x)$. The second formula is defined by:

$$[B_1 = f(x,B_1)] \wedge [B_2 = f(x,B_2)] \ldots [B_n = f(x,B_l)],$$

where each $B_i \in D$. We abbreviate this formula by $D(x)$. The sequence $C(x) \to D(x)$ is called a *decision rule induced by x*.

For any formula α, let $|\alpha|$ denote the set of objects satisfying α, and for any finite set X, let $\#(X)$ denote the number of elements of X. Then, with every decision rule $C(x) \to D(x)$, where both $|C(x)|$ and $|D(x)|$ are non empty, rough set theory associates the *certainty factor* of the decision rule, denoted by $cer_x(C;D)$ and defined as follows:

$$cer_x(C;D) = \frac{\#(|C(x)| \cap |D(x)|)}{\#|C(x)|}$$

If $cer_x(C;D) = 1$, then $C(x) \to D(x)$ will be called a certain decision rule; if $0 < cer_x(C;D) < 1$, then the decision rule will called an uncertain decision rule. Another important factor is a coverage of the decision rule, denoted $cov_x(C;D)$ and defined as:

$$cov_x(C;D) = \frac{\#(|C(x)| \cap |D(x)|)}{\#|D(x)|}$$

If $C(x) \rightarrow D(x)$ is a decision rule then

$$\bigcup_{y \in |D(x)|} = \{|C(y)| : |C(y)| \subseteq |D(x)|\}$$

is the lower approximation of the decision class $|D(x)|$, whereas

$$\bigcup_{y \in |D(x)|} = \{|C(y)| : |C(y)| \cap |D(x)| \neq \emptyset\}$$

is the upper approximation of $|D(x)|$.

24.4 Supervaluationist Semantics

In this section, we discuss in some detail supervaluationism. Firstly, we present an informal (philosophical) introduction and then define a formal account. Our aim is to show affinities between supervaluationism and rough set theory in ways of dealing with vague predicates.

Fine, the leading supervaluationist philosopher, introduced the notion of a specification space as a formal tool to deal with vagueness [2]. Such a space is a set of points (i.e., specification-points) at which some or all sentences of a given language L are assigned truth-values. Some specification-points may be complete, yet some points may be incomplete. The former points correspond to ways of making all vague terms completely precise; the latter correspond to ways where some or all expressions are left vague and some sentences lack a truth-value. A specification space is additionally equipped with a binary extends relation, which should satisfy certain important restrictions. The stability requirement states that the sentences which are true (false) at a specification-point must remain unchanged at any point which extends it. The requirement of fidelity states that a complete point is in actual fact a classical valuation. Another requirement is the completability condition: every specification-point can be extended to a complete specification-point. The stability requires that if a sentence is true at a partial specification-point, then it must be true at all complete points that extend this point. Finally, every specification space must have a base point; all other points must extend the base point.

Supervaluationism, as it is often put, *quantifies over complete and admissible specifications*. Thus, truth (or better still, super-truth) is truth at the base-point, and hence, at all complete specification-points which extend the base-point. Surprisingly, this model preserves many classical tautologies, yet it gives new semantics to logical connectives. Let us once again consider Adam whose BMI = 26. "Adam is obese or Adam is not obese" is always true, that is, true at any complete point, yet both "Adam is obese" and "Adam is not obese" lack a truth (super-truth) value. Thus, although neither of two sentences is true, their disjunction may be true.

These philosophical requirements can be formalized as follows [7]:

Definition 24.7 (Partial Model). Given a first-order language L, a partial model M for L is an ordered pair $\langle U, I\rangle$, where U, the domain of discourse, is a non-empty set, and I is a function assigning

- to each name of L a member of U;
- to each n-place function symbol of L an n-place function on D;
- to some, all or none of the propositional variables of L a member of $\{t, f\}$ so that I restricted to the propositional variables is a partial function;
- to each n-place relation symbol R of L an ordered pair $I(R) = \langle I^+(R), I^-(R)\rangle$ with $I^+(R) \subseteq U^n$, $I^-(R) \subseteq U^n$ and $I^+(R) \cap I^-(R) = \emptyset$. $I^+(R)$ and $I^-(R)$ are the extension and the anti-extension of R.

Partial model is actually a base point. Other models to be considered are precisifications of this base model. As one can easily see, the extension and anti-extension are very similar to the disjoint representation of rough sets.

Definition 24.8 (Precisification). A partial model $M' = \langle U, I'\rangle$ is a precisification of $M = \langle U, I\rangle$ (in symbols, $M \leq M$) iff

- $I'(c) = I(c)$ for each name or function symbol c;
- for each propositional variable p, if $I(p)$ is defined then $I'(p) = I(p)$;
- $I^+(R) \subseteq I'^+(R)$ and $I^-(R) \subseteq I'^-(R)$ for each relational symbol.

A partial model $M = \langle U, I\rangle$ is classical iff both $I(p)$ is defined for each propositional variable and $I^+(R) \cup I^-(R) = U^n$ for each n-place relation symbol R.

Precisifications allow one to make a partial model more precise by adding new objects to an extension $I^+(R)$ of a relational symbol R of L.

Now, let us come back to an information system. Each atomic formula $[A = val]$ of $\mathcal{L}_{Desc}$ can be regarded as a predicate $[A = val](x)$, and thus, an information system gives rise to first-order language (consisting of only monadic predicates). Of course, in the case of a Boolean attribute A, one can abbreviate $[A = Yes](x)$ as $[A](x)$. Then, a model is actually a function from first-order $\mathcal{L}_{Desc}$ to an information system. Suppose, that we are dealing with a vague predicate "obese"; Fig. 24.1 presents a partial model for this predicate.

Segment	BMI	Obese
1	$0 < BMI \leq 15$	No
2	$15 < BMI \leq 20$	No
3	$20 < BMI \leq 24$	
4	$24 < BMI \leq 27$	
5	$27 < BMI \leq 30$	
5	$30 < BMI \leq 35$	
6	$35 < BMI \leq 40$	Yes
7	$40 < BMI$	Yes

Fig. 24.1. A data table (i.e. information system) corresponding to a partial model for "obese"

This model is regarded as a base-point, where

$$I^{+}([Obese]) = \{6,7\} \text{ and } I^{-}([Obese]) = \{1,2\}.$$

The complete precisifications of this partial model are presented in Fig. 24.2. Of course, it is only part of possible precisifications. In rough set theory, one could

Segment	BMI	Obese
1	$0 < BMI \leq 15$	No
2	$15 < BMI \leq 20$	No
3	$20 < BMI \leq 24$	No
4	$24 < BMI \leq 27$	Yes
5	$27 < BMI \leq 30$	Yes
6	$30 < BMI \leq 35$	Yes
7	$35 < BMI \leq 40$	Yes
8	$40 < BMI$	Yes

Segment	BMI	Obese
1	$0 < BMI \leq 15$	No
2	$15 < BMI \leq 20$	No
3	$20 < BMI \leq 24$	No
4	$24 < BMI \leq 27$	No
5	$27 < BMI \leq 30$	Yes
6	$30 < BMI \leq 35$	Yes
7	$35 < BMI \leq 40$	Yes
8	$40 < BMI$	Yes

Segment	BMI	Obese
1	$0 < BMI \leq 15$	No
2	$15 < BMI \leq 20$	No
3	$20 < BMI \leq 24$	No
4	$24 < BMI \leq 27$	No
5	$27 < BMI \leq 30$	No
6	$30 < BMI \leq 35$	Yes
7	$35 < BMI \leq 40$	Yes
8	$40 < BMI$	Yes

Fig. 24.2. Complete precisifications of "obese"

merge these precisifications into a single complete information system; such a system for "obese" is depicted by Fig. 24.3. As usual, $Obese \in Att$ is regarded as a

Segment	BMI	Obese
1	$0 < BMI < 15$	No
2	$15 < BMI < 20$	No
3	$20 < BMI < 24$	No
4	$24 < BMI < 27$	Yes
4	$24 < BMI < 27$	No
5	$27 < BMI < 30$	Yes
5	$27 < BMI < 30$	No
6	$30 < BMI < 35$	Yes
7	$35 < BMI < 40$	Yes
8	$40 < BMI$	Yes

Fig. 24.3. Information system representing precisifications of "obese"

decision attribute, whereas $BMI \in Att$ is a conditional attribute. Of course, the indiscernibility relation is computed with respect to conditional attributes (i.e., *BMI*).

The supervaluationist semantics emphasizes that there are many precisifications (of any vague predicate) and all of them should be take into account: *[T]here are many equally good precisifications corresponding to different positive and negative extensions. According to supervaluationism, by taking account of all precisifications we can provide the logic and semantics of vague language* [6]. Suppose that Adam's BMI = 26, that is, Adam is a borderline case of obesity. As one can see, there is a precisification which places him in positive extension of "obese", but others place him in negative extension (i.e., anti-extension). Thus, "Adam is obese" is neither true on all precisifications nor false on all of them; hence, it counts as neither true nor false. Furthermore, different precisifications draw boundaries to the obese people at different values of BMI. And, as Fodor and Lepore observe, actually there are infinitely many such precisifications and, in consequence, for no value n, it is the case that on all precisifications only people whose BMI is equal or greater than n are counted as obese. In this way, supervaluationism is able to model higher-order vagueness: the boundary region is vague. But, as Keefe argues, even in the finite case supervaluationism is able to maintain higher-order vagueness: *super-truth is vague because it is a matter of truth in all admissible specifications and 'all admissible specifications' is itself vague* [6]. Thus, since the metalanguage used by supervaluationists is vague, vagueness must occur also on object level.

However, in rough set theory, a complete information system must be well-defined and finite, and, as a result, we can actually compute such a value n. Thus, and it is no surprise, the boundary region of "obese" in the theory of rough sets is strict. Let $|\alpha|$ denote the set of all objects satisfying a formula $|\alpha|$ of $\mathcal{L}_{Desc}$. Then

$$\underline{|[Obese = Yes]|} = \{6,7,8\}$$

$$\overline{|[Obese = Yes]|} = \{4,5,6,7,8\}$$

$$Ext(|[Obese = Yes]|) = \{1,2,3\}$$

$$Bnd(|[Obese = Yes]|) = \{4,5\}$$

Anyway, according to supervaluationist semantics if $n \in \underline{|[Obese = Yes]|}$, then "$n$ is obese" is true (supertrue); if $n \in Ext(|[Obese = Yes]|)$, then "$n$ is obese" is false; and if $n \in Bnd(|[Obese = Yes]|)$, then "$n$ is obese" is neither true nor false. Thus, the supervaluationism on the level of atomic sentences coincides with rough set theory.

Let us consider the information system at issue as a decision table: $C = \{BMI\}$, $D = \{Obese\}$. Then for every $x \in \underline{|[Obese = Yes]|}$, it holds that

$$cer_x(C;D) = \frac{\#(|C(x)| \cap |D(x)|)}{\#|C(x)|} = 1,$$

for every $x \in \overline{|[Obese = Yes]|}$, it holds that

$$cer_x(C;D) = \frac{\#(|C(x)| \cap |D(x)|)}{\#|C(x)|} > 0,$$

and for every $x \in Ext(|[Obese = Yes]|)$

$$cer_x(C;D) = \frac{\#(|C(x)| \cap |D(x)|)}{\#|C(x)|} = 0.$$

Of course, logics stemming from both approaches are different. Yet, as is well known, the lower approximation operator and the upper approximation operator are actually $\Box$ and $\Diamond$ of modal system S5, respectively.

Definition 24.9 (Modal Language). Let be given a set of proposition letters Φ whose elements are usually denoted p, q, r, and so on, and a unary modal operator $\Diamond$ ('diamond'). The well-formed formulas of the basic modal language are given by the rule:

$$\alpha ::= p|\neg\alpha|\alpha \vee \beta|\Diamond\alpha,$$

where p ranges over elements of Φ. The set of well-formed formulas is denoted by $\mathcal{F}$.

As usual, we also make use of the classical abbreviations for conjunction $\wedge$, implication $\rightarrow$, bi-implication $\leftrightarrow$, the constant true $\top$, and the modal operator box $\Box$.

Definition 24.10 (Kripke Model). A Kripke model is a triple $\langle W, R, v\rangle$, where W is a non-empty set of worlds, R is a binary relation on W, and $v : W \times \Phi \rightarrow \{0,1\}$ is function assigning to each pair (w,p) a truth value. We usually write $v_w(p) = 1$ (or $v_w(p) = 1$), which is read as *at the world w, p is true (or false)*. The function v is extended on every formula $\alpha \in \mathcal{F}$ in the standard way:

- $v_w(\neg\alpha) = 1$ if $v_w(\alpha) = 0$, and 0 otherwise,
- $v_w(\alpha \wedge \beta) = 1$ if $v_w(\alpha) = 1$ and $v_w(\beta) = 1$, and 0 otherwise,
- $v_w(\Diamond\alpha) = 1$ if, for some $w' \in W$ such that wRw', $v_{w'}(\alpha) = 1$, and 0 otherwise,
- $v_w(\Box\alpha) = 1$ if, for all $w' \in W$ such that wRw', $v_{w'}(\alpha) = 1$, and 0 otherwise.

Every Kripke model induces $Prop : \mathcal{F} \rightarrow \mathcal{P}(W)$, defined by

$$Prop(\alpha) = \{w \in W : v_w(\alpha) = 1\}.$$

Definition 24.11 (Semantic Consequence). Let $\Sigma \subseteq \mathcal{F}$ and $\alpha \in \mathcal{F}$:

- $\Sigma \models \alpha$ iff for all Kripke models $\langle W, R, v\rangle$ and all $w \in W$, if $v_w(\beta) = 1$ for all $\beta \in \Sigma$, then $v_w(\alpha) = 1$;
- $\models \alpha$ iff $\emptyset \models \alpha$.

Proposition 24.1. *Every approximation space (U,E) and a map $\phi : \Phi \rightarrow \mathcal{P}(U)$ give rise to a Kripke model $\langle U, E, v\rangle$ of S5, where $v_a(p) = 1$ iff $a \in \phi(p)$. Furthermore*

$$Prop(\Box\alpha) = \underline{Prop(\alpha)} \quad and \quad Prop(\Diamond\alpha) = \overline{Prop(\alpha)}.$$

Now, coming back to supervaluationism, we can obtain the following characterization [7]:

Proposition 24.2

$$\Sigma \models \alpha \quad \textit{iff} \quad \Box\Sigma \models_{S5} \Box\alpha,$$

where $\models$ *is a supervaluationist consequence operator, whereas* $\models_{S5}$ *is an S5 consequence operator, and*

$$\Box\Sigma = \{\Box\alpha : \alpha \in \Sigma\}.$$

The reader interested in formal properties of supervaluation-based consequence should consult [7]. Here, in the chapter, we would like to focus on how rough set theory is closely related to supervaluationist approach to vagueness.

Philosophically, the rough set approach is weaker than supervaluationism because the boundary region is crisp. On the other hand, it allows one to exactly define borderline cases. As Weiner observes: *as sometimes happens in such research, the decision is made to exclude borderline cases from the study. [...] For obvious reasons – the exclusion of borderline cases requires two sharp distinctions: a distinction between those who are obese and those who are borderline-obese and a distinction between those who are borderline-obese, and those who are not obese* [19]. Please observe, that it is exactly what we have in rough set theory. Following the line offered by Weiner, one could say that rough set theory is a formalization of scientific version of supervaluationism. Or better still, since it is Frege's philosophy which is the closet to everyday scientific practice, rough sets may be regarded as a step from supervaluationism toward Frege.

In the next section, we discuss in detail such a Fregean variant of rough sets.

24.5 Vagueness in Science

In this section we consider a scientific approach to vague predicates. We focus our attention on the usage of the term "obesity" in medical sciences. Of course, as noted above, one can regard the term "obesity" used in medical research as a non-English technical term which is not relevant to philosophical discussions about vagueness [3]. However, as Weiner observed, this term was not introduced to describe some well-understood and determinate property. On the contrary, among obesity researches, it is regarded as a vague term *not appreciably different from the everyday word 'fat'* [19]. Thus, since we seriously consider research results about, *for example*, heart disease, we should also seriously consider what researches do with vague predicates.

As cited in the introductory section, *obesity is a major contributor to serious health conditions in children and adults, including cardiovascular disease*. In order to determine whether this hypothesis is true, a scientist would take a group of individuals who suffer from cardiovascular disease and a group of individuals who do not (a control group), and then she would compute the proportion of each group that

is obese. The problem is that "obese" is a vague term and therefore such computation would seem impossible. Weiner writes:

The solution, in obesity research, is to use measurable index of weight relative to height – typically, body mass index (BMI). And the researcher will decide, antecedent to beginning the study, on sharp cut-off (say, BMI $>$ 30) between the obese and non-obese. It is essential to such an investigation to draw a line.

Thus, scientists use one single classical precisification (i.e. BMI $>$ 30). Their aim is defined by a theory that some diseases may be related to obesity. If there is a particular value of BMI where negative effects on health begin, then a scientist may conclude that it is a line between obese and non-obese and that the hypothesis is true. If there is no such a value, then the hypothesis is false. This methodology is quite different form the supervaluationist approach. Suppose that BMI $>$ 31 is also an admissible precisification of "obese". Suppose further that actually only people whose BMI is greater than 31 suffer from cardiovascular disease. Then according to supervaluationism the hypothesis is neither true nor false: it is false under the precisification BMI $>$ 30 and true under BMI $>$ 31. But for a scientist such an answer is wrong: the hypothesis must eventually be true or false. That is why the single classical precisification must be used; in this case, it would be BMI $>$ 31.

Segment	BMI	Heart Disease	Number
1	$0 < BMI \leq 15$	No	100
2	$15 < BMI \leq 20$	No	100
3	$20 < BMI \leq 24$	No	100
4	$24 < BMI \leq 27$	Yes	5
4	$24 < BMI \leq 27$	No	95
5	$27 < BMI \leq 30$	Yes	95
5	$27 < BMI \leq 30$	No	5
6	$30 < BMI \leq 35$	Yes	100
7	$35 < BMI \leq 40$	Yes	100
8	$40 < BMI$	Yes	100

Fig. 24.4. Information system representing precisifications of "obese"

Interestingly, the rough set approach to data is close to the scientific methodology, despite the fact that the concept of a rough set is actually close to supervaluationism. Of course, as long as we start with an approximation space, we must end with the borderline cases. However, most of Pawlak's papers start with a decision information system, where we have distinguished condition and decision attributes and our aim is actually to approximate decision attributes in terms of conditional ones. To do so we must generate decision rules and thus, as in empirical science, we must assume a hypothesis that some conditional attributes are related to a given decision attribute. In consequence, as "obese" is defined by means of BMI and cardiovascular disease, similarly, in rough set theory, a decision attribute is defined in terms of conditional attributes. However, there is a substantial difference between rough set

theory and empirical sciences in treating these hypothesises: in rough set theory we always assume that there is a very strong correspondence between the conditional attributes and the decision attribute (that is why decision rules presented in papers have always very high quality), whereas in science the correspondence is viewed in a theoretical context (*e.g.* a rule that a smoking person suffers from lungs cancer has a weak certainty factor, but this factor is much higher than the certainty of the rule that a non-smoking person suffers lungs cancer) – we shall return to this issue soon.

Consider a modified version of our previous information system; see Fig. 24.4. Suppose that in each range of BMI we examined 100 patients whether they suffer from cardiovascular disease. The results are presented in the column Number. Then, as once can easy compute:

$$cer_x([BMI \leq 24]; [HeartDisease = No]) = 1,$$

$$cer_x([BMI \leq 27]; [HeartDisease = No]) = \frac{95}{100},$$

$$cer_x([27 < BMI]; [HeartDisease = Yes]) = \frac{95}{100},$$

$$cer_x([30 < BMI]; [HeartDisease = Yes]) = 1.$$

A scientist would say that obesity starts with BMI greater than 27 and that hypothesis concerning the relationship between obesity and cardiovascular disease is true. However, from the standpoint rough set theory (based on approximation spaces) things look a bit different: the lower approximation of $[27 < BMI] \rightarrow [HeartDisease = Yes]$ is $\{6,7,8\}$, the upper approximation is $\{4,5,6,7,8\}$; thus the boundary region is $\{4,5\}$. In consequence, we obtain two different classifications; the comparison of these two classifications is presented by Fig. 24.5 and Fig. 24.6.

Segment	BMI	Heart Disease	Number	Obese
1	$0 < BMI \leq 15$	No	100	No
2	$15 < BMI \leq 20$	No	100	No
3	$20 < BMI \leq 24$	No	100	No
4	$24 < BMI \leq 27$	Yes	5	No
4	$24 < BMI \leq 27$	No	95	No
5	$27 < BMI \leq 30$	Yes	95	Yes
5	$27 < BMI \leq 30$	No	5	Yes
6	$30 < BMI \leq 35$	Yes	100	Yes
7	$35 < BMI \leq 40$	Yes	100	Yes
8	$40 < BMI$	Yes	100	Yes

Fig. 24.5. Scientific classification of obese

As one can see, the answer based on rough set theory would be that "obese" for sure starts with BMI > 30, and "obese" might begin with BMI > 24. But intuitively, the answer given by a scientist, that is, "obese" starts with BMI > 27, seems to be better:

Segment	BMI	Heart Disease	Number	Obese
1	$0 < BMI \leq 15$	No	100	No
2	$15 < BMI \leq 20$	No	100	No
3	$20 < BMI \leq 24$	No	100	No
4	$24 < BMI \leq 27$	Yes	5	?
4	$24 < BMI \leq 27$	No	95	?
5	$27 < BMI \leq 30$	Yes	95	?
5	$27 < BMI \leq 30$	No	5	?
6	$30 < BMI \leq 35$	Yes	100	Yes
7	$35 < BMI \leq 40$	Yes	100	Yes
8	$40 < BMI$	Yes	100	Yes

Fig. 24.6. Rough set classification of obese

in actual fact *almost all* people with BMI greater than 27 suffer from cardiovascular disease. But, when we begin the analysis with decision rules and their quality (as Pawlak in his papers) then the resulting classification will change. Let us consider decision rules of the type

$$[the\ \ range\ \ of\ \ BMI] \rightarrow [HeartDisease = Yes]$$

and compute their certainty for each range of BMI. Since the certainty of the above rule is high for BMI > 27, we would count as obese all persons whose BMI is greater than 27, see Fig. 24.7. Thus, on the level of decision rules we would obtain exactly the same results as in empirical sciences; compare Fig. 24.5 and Fig. 24.7. Therefore decision rules may be seen as a scientific core of the rough set methodology.

Segment	BMI	Heart Disease	Number	Cert	Obese
1	$0 < BMI \leq 15$	No	100	0	No
2	$15 < BMI \leq 20$	No	100	0	No
3	$20 < BMI \leq 24$	No	100	0	No
4	$24 < BMI \leq 27$	Yes	5	0.05	No
4	$24 < BMI \leq 27$	No	95	–	No
5	$27 < BMI \leq 30$	Yes	95	0.95	Yes
5	$27 < BMI \leq 30$	No	5	–	Yes
6	$30 < BMI \leq 35$	Yes	100	1	Yes
7	$35 < BMI \leq 40$	Yes	100	1	Yes
8	$40 < BMI$	Yes	100	1	Yes

Fig. 24.7. Rough set classification of obese based on certainty of decision rules

However, this coincidence of the rough set classification based on the quality of decision rules and the scientific classification does not mean that there is no difference between these two approaches. In order to better explain differences between the rough set methodology and the scientific one, let us briefly consider a few more

examples. Suppose now that in all rows equal or higher than 4 we have got 30 patients suffering from a heart disease and 70 patients who do not. Then according to the (classical) rough set analysis the lower approximation is empty, and the upper approximation is $\{4,5,6,7,8\}$. This time, the rough set analysis based on decision rules would give the same answer as the classical approach: the highest quality of the rules is 0.3, which seems very low. However, a scientist would say that obesity starts with BMI greater than 24 and all people with BMI less than this value are non-obese. The reason is that for this value of BMI negative effects on health begin and as in the case of lungs cancer and smoking, it is the difference between the quality of decision rules computed for different values of BMI which is important. As a result, the scientific answer would still bring dichotomic division on obese and non-obese. More interestingly, given that in all rows the number of healthy persons and the number of ill persons are similar, we would have a boundary region $\{1,2,3,4,5,6,7,8\}$ in both classical and decision rule based rough set analysis, but a scientist would stay with a previous (default) definition of obesity (i.e. BMI $>$ 30) and say that this time the hypothesis is false. Of course, in rough set theory we would also say that on the basis of such data we cannot build a good classifier. Anyway, the main methodological difference is that in science we start with a default definition of obesity (*e.g.* BMI > 30) and a null hypothesis (*e.g.* that there is no relationship between obesity and heart a heart disease). Then a scientist checks whether collected data reject the null hypothesis. If the hypothesis can be rejected under a slight modification of the default definition of obesity, then the term is redefined, and the alternative hypothesis is accepted. The reason for this procedure is that one cannot prove a hypothesis, but only reject it; that is why one must have two hypothesizes: a null hypothesis to reject, and an alternative hypothesis to corroborate. On the other hand, rough set theory attempts to obtain all pieces of information from data, and that is why in the last scenario, this theory would give no answer who is obese (all segments belong to the boundary region).

The first thing which we can learn from the above examples is that in some cases the quality of decision rules must not be regarded as absolute, but rather as relative to the context of application. Of course, if one deals with crisp concepts which have borderline cases due to the incompleteness of one's knowledge, then the standard rough set analysis will usually be appropriate. But when one deals with a vague concept such as "obese", then one must read the quality of decision rules in a context. As above, sometimes the quality 0.3 is good enough to define a decision attribute precisely. Here, the change of quality is more important than its numerical values. Furthermore, the example shown by Fig. 7 demonstrates that sometimes the borderline area actually makes things look worse than it is the case, and that by making the vague concept precise we may obtain much better (adequate) classification than the original based on approximation operators.

The second thing suggested by the above examples is to assume a definition of a given vague concept and some default (null) hypothesis, and then, if necessary, to change this definition so as to make it better fit to collected data. Concerning the former, since for a given non-empty set X its upper approximation $\overline{X}$ is always non-empty, we could regard it as a default crisp definition of X. Then our aim would

be to modify this definition while maintaining its crispness. Concerning the latter, this suggestion is against the basic methodological assumptions underlying rough set theory: hypothesis (decision rules) must come from data (no a priori rules). Yet, we can try to obtain a posteriori hypothesizes which would allow us to modify $\overline{X}$.

The procedure would be as follow. Firstly, we would compute $cer_x(C;D)$ for every $x \in \overline{X}$. Given that, we would define a tolerance relation $T \subseteq U \times U$ as it is done in near set theory [13, 14]:

$$T_\varepsilon = \{(x,y) \in U \times U : abs(cer_x(C;D) - cer_y(C;D)) \leq \varepsilon\},$$

where $abs(n)$ denotes the absolute value of n.

Definition 24.12 (Preclass, Class). Let T be a tolerance relation. Then $X \subseteq U$ is called a *preclass* of T iff, for all $x, y \in X$, it holds that xTy. A preclass X is called a *class*, if it is a maximal preclass.

Now let us consider a decision table with a single – for simplicity – decision attribute $d \in D$ and its distinguished value $v_d \in Val_d$ (*e.g.* *Obese* and *Yes*). As said above, we would like to replace a rough set $(\underline{|[d = val_d]|}, \overline{|[d = val_d]|})$ (*e.g.* $(\underline{|[Obese = Yes]|}, \overline{|[Obese = Yes]|})$ which represents a vague predicate "obese") by a crisp set. To this end, we can take a set of classes (or preclasses) of

$$T_\varepsilon \cap \overline{|[d = val_d]|} \times \overline{|[d = val_d]|},$$

$$e.g. \quad T_\varepsilon \cap \overline{|[Obese = Yes]|} \times \overline{|[Obese = Yes]|}.$$

For each class (preclass) W, we could take its set of decision rules

$$\{C(x) \rightarrow D(x) : x \in W\}$$

and compute the average certainty defined in one way or the other, *for example* arithmetic mean. Finally, it suffices to choose a class (preclass) Z whose average certainty is the highest and regard Z as a new precisification of d. In this way, we can made a dichotomic division on d (i.e., Z) and non-d (i.e., $U \setminus Z$). Of course, in this method we can use any form of decision rules (not necessarily complete rules defined for all conditional attributes): *for example,* $F(x) \rightarrow D(x)$, $F \subseteq C$. Needless to say, it is only one of many possible methods of how we can convert d into a crisp concept. We have chosen the framework of near sets due to the parameter ε which can be tuned so as to make the modification of $|[d = val_d]|$ fit better to data.

Of course, these remarks need not be treated as a criticism of rough set theory but as an offer how we can modify it. As Weiner puts it [19]:

I offer these arguments, not as an attempt to urge the rejection of supervaluationism (or any other account of the semantics of vague predicates) but as a challenge – to come up with an account that does not do violence to our views about what we have learned from empirical research.

In our opinion, rough set theory is a close cousin of supervaluationism which is a bit more scientific. The methodology based on decision rules actually is very

close to what is done is science. Yet, we can make it even closer, and on the basis of the quality of decision rules, try to define a decision attribute as a crisp concept. Actually, the quality measures of decision rules considered by Pawlak seem to be a good tool to make the final classification precise.

Summing up this section, let us come back to Frege. Weiner asks: *Why should the laws of truth require sharp boundaries?* Then she answers: *Because the realm of truth is the realm of science*, and that is why Frege with his demands concerning sharp boundaries of concepts should be regarded *not as a philosopher of language, but as a philosopher of science* [18]. As a matter of fact, Frege is quite often cited in papers about rough sets, *for example* [12]; but the true application of his philosophy to rough sets consists on making decision attributes crisp.

24.6 Conclusions

Rough set theory is often presented as a mathematical tool to deal with vagueness in data, yet there is lack of information how rough sets are really related to present approaches to vagueness. In the chapter, we have presented rough set theory against the background of philosophical discussions about vagueness. We have chosen the recent discussion concerned with vagueness in empirical sciences. In our opinion, the rough set approach to vagueness is closely related to supervaluationism, but it is a bit shifted towards the scientific methodology. The final step towards this direction would be an attempt to make – on the basis of the quality of decision rules – a decision attribute a crisp concept. In this way, we would obtain a true mathematization of Frege's approach to vague predicates.

Acknowledgements. I am indebted to the anonymous referee for the valuable comments and suggestions. The research has been supported by the grant NN516 077837 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Bazan, J., Skowron, A., Świniarski, R.: Rough Sets and Vague Concept Approximation: From Sample Approximation to Adaptive Learning. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 39–62. Springer, Heidelberg (2006)
2. Fine, K.: Vagueness, truth and logic. Synthese 30(3-4), 265–300 (1975)
3. Fodor, J.A., Lepore, E.: What cannot be evaluated cannot be evaluated and it cannot be superevaluated either. The Journal of Philosophy 93(10), 516–535 (1996)
4. Frege, G.: The Basic Laws of Arithmetic: Exposition of the System. In: Furth, M. (ed.) Trans., vol. I. University of California Press, Los Angeles (1964)
5. Frege, G.: Posthumous Writings. In: Kambartel, H., Kaulbach (eds.) Trans. Long and White. Basil Blackwell, Oxford (1979)

6. Keefe, R.: Theories of Vagueness. Cambridge Studies in Philosophy. Cambridge University Press (2000)
7. Kremer, P., Kremer, M.: Some supervaluation-based consequence relations. Journal of Philosophical Logic 32(3), 225–244 (2003)
8. Marek, W., Pawlak, Z.: Information storage and retrieval systems, mathematical foundations. Theoretical Computer Science 1(4), 331–354 (1976)
9. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sci. 11, 341–356 (1982)
10. Pawlak, Z.: Rough logic. Bull. Polish Acad. Sc (Tech. Sc.) 35(5-6), 253–258 (1987)
11. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publisher, Dordrecht (1991)
12. Pawlak, Z.: An inquiry into vagueness and uncertainty. Institute of Computer Science Report 29/94. Warsaw University of Technology (1994)
13. Peters, J.F.: Near sets. Special theory about nearness of objects. Fundamenta Informaticae 75, 407–433 (2007)
14. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79, 497–512 (2007)
15. Skowron, A.: Rough sets and vague concepts. Fundamenta Informaticae 64(1-4), 417–431 (2005)
16. Skowron, A., Świniarski, R.W.: Rough Sets and Higher Order Vagueness. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 33–42. Springer, Heidelberg (2005)
17. Weiner, J.: Frege in Perspective. Cornell University Press, Ithaca (1990)
18. Weiner, J.: Has Frege a philosophy of language? In: Tait, W. (ed.) Early Analytic Philosophy: Essays in Honor of Leonard Linski. Open Court Press (1997)
19. Weiner, J.: Science and semantics: the case of vagueness and supervaluation. Pacific Philosophical Quarterly 88(3), 355–374 (2007)

Index